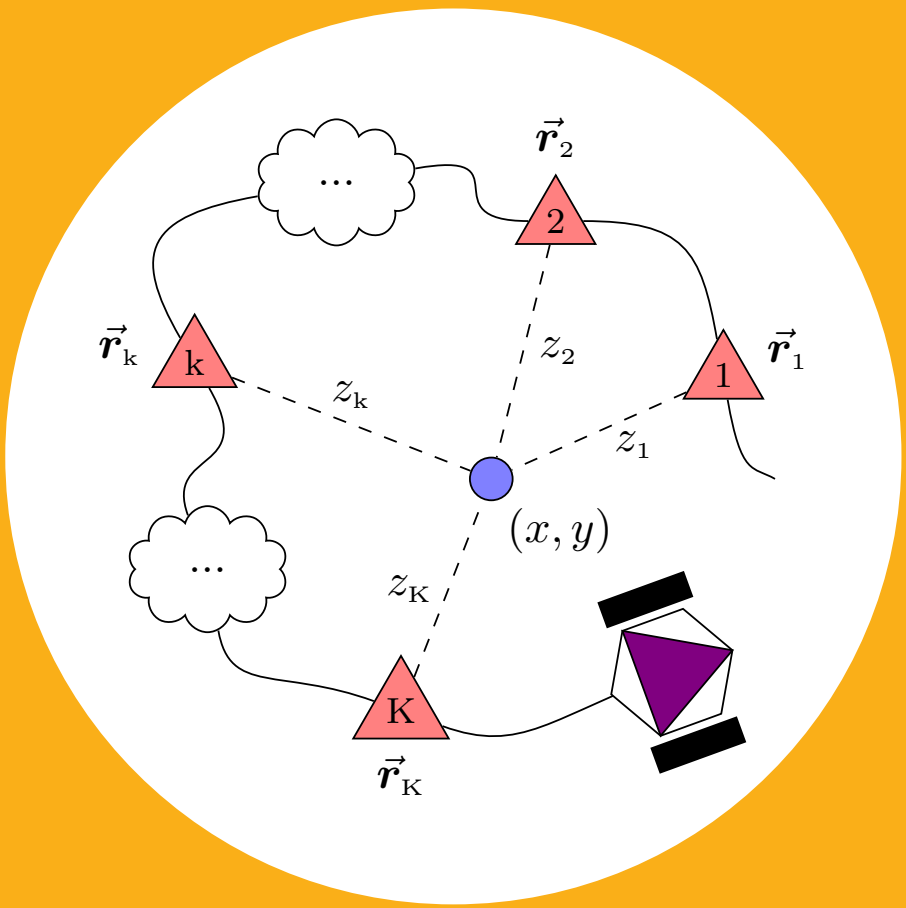
Department of Electrical Engineering and Automation

Localization in Wireless Sensor Networks Using a Mobile Robot

José María Vallet García



Aalto University

DOCTORAL DISSERTATIONS

Aalto University publication series
DOCTORAL DISSERTATIONS 56/2016

Localization in Wireless Sensor Networks Using a Mobile Robot

José María Vallet García

A doctoral dissertation completed for the degree of Doctor of Science (Technology) to be defended, with the permission of the Aalto University School of Electrical Engineering, at a public examination held at the lecture hall AS1 in the TUAS building (Otaniementie 17, Espoo, Finland) on the 15th of April 2016 at 12 o'clock noon.

Aalto University
School of Electrical Engineering
Department of Electrical Engineering and Automation

Supervising professors
Professor Ville Kyrki
Professor Aarne Halme

Preliminary examiners
Professor Henk Wymeersch, Chalmers University of Technology, Sweden

Professor Gonzalo Seco-Granados, Universidad Autónoma de Barcelona, Spain

Opponent
Professor Fredrik Gustafsson, Linköping University, Sweden

Aalto University publication series
DOCTORAL DISSERTATIONS 56/2016

ISBN 978-952-60-6725-4 (printed)
ISBN 978-952-60-6726-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)
http://urn.fi/URN:ISBN:978-952-60-6726-1

Corrected version / second edition

Unigrafia Oy
Helsinki 2016

Finland

Aalto University

Abstract

Aalto University, P.O. Box 11000, FI-00076 Aalto www.aalto.fi

Author
José María Vallet García

Name of the doctoral dissertation
Localization in Wireless Sensor Networks Using a Mobile Robot

Publisher School of Electrical Engineering

Unit Department of Electrical Engineering and Automation

Series Aalto University publication series DOCTORAL DISSERTATIONS 56/2016

Field of research Automation Technology

Manuscript submitted 10 September 2015 **Date of the defence** 15 April 2016

Permission to publish granted (date) 28 January 2016 **Language** English

☒ **Monograph** ☐ **Article dissertation** ☐ **Essay dissertation**

Abstract

This thesis presents studies and methods relevant to the problem of localization in wireless sensor networks (WSN), with the ultimate goal of producing practical solutions that can be used in real adhoc deployments. The motivational sample application type is emergency and rescue operations, which are characterized by the lack of a pre-installed infrastructure and on-site training data.

The base scenario is an unexplored environment in which the nodes of a WSN are distributed in random unknown positions. A robot capable of simultaneous localization and mapping is used as a mobile beacon, with the help of which the nodes' position can be estimated accurately using received signal strength (RSS) measurements. Using data collected in three different environments, we demonstrate sub-metre accuracy for some of the proposed methods, part of which are self-adaptive and can cope with changes in the environment.

The localization algorithms are based on least squares (LS) and maximum likelihood (ML) estimation relying on parametric measurement models. In order to obtain a realistic confidence indicator on the estimates, special attention is paid to the calculation of their covariance. The work presented includes studies on the variability of the log-normal model parameters typically observed in WSNs, the sensitivity of ML position estimators due to this variability, the over-confidence of the estimates under the assumption of identically and independently distributed errors, the spatial autocorrelation of the RSS and the usage of concentrated log-likelihoods to jointly estimate position and model parameters while solving identifiability and convergence issues using regularization.

In addition to ML, three families of linear position estimators are studied and evaluated, two of which are new. Unlike non-linear methods, they do not require initial estimates. Two of them have closed analytical forms, and therefore are computationally efficient. The third is iterative, and it has demonstrated an excellent performance comparable to ML in our experiments.

All in all, besides contributing to the field of localization, this work represents a small step towards understanding and leveraging the potential benefits of using mobile robots as assistive localization devices.

Keywords Localization, wireless sensor networks, mobile robot, beacons, received signal strength, channel modelling, spatial autocorrelation, joint localization and model parameters estimation, regularization.

ISBN (printed) 978-952-60-6725-4 **ISBN (pdf)** 978-952-60-6726-1

ISSN-L 1799-4934 **ISSN (printed)** 1799-4934 **ISSN (pdf)** 1799-4942

Location of publisher Helsinki **Location of printing** Helsinki **Year** 2016

Pages 302 **urn** http://urn.fi/URN:ISBN:978-952-60-6726-1

Preface

This thesis is the result of years of dedicated research that begun to take off under the framework of the WISM2 project (11/2010 - 02/2013). WISM stands for *wireless sensor systems in indoor situation modelling*, and its aim was to use wireless sensor networks (WSNs) and other technologies to build and enhance awareness in emergency and rescue operations. This requires the capability of locating the nodes of the network in previously unexplored environments and with no pre-installed infrastructure. I was invited to join the project as a localization specialist to solve that particular problem. Being part of a robotics lab, the idea of using a robot as a mobile beacon arose naturally and almost instantaneously. The result was an interesting, focused and challenging but tractable problem suitable for a doctoral dissertation that appears in many practical scenarios, emphasizing the potential applicability of the research outcomes.

It took me several years of determined work to arrive to the point at which this book would be ready for submission. Besides personal work, the research activity usually requires a proper context that facilitates the work and symbiotic cooperation with other researchers and colleagues. In this respect, I could not conceive of my book in its present form without those that conformed this context, and to whom I am deeply thankful.

First, I would like to thank my supervisors, Professors Ville Kyrki and Aarne Halme, who provided me with inestimable insights that helped me shape my research and its presentation. I feel privileged after having enjoyed technical discussions with Professor Kyrki and his exemplary and detailed feedback. Professor Halme was also the leader of the *generic intelligent machines* (GIM) centre of excellence, the main funding instrument for my research, and I am especially grateful to him for his continuous support and trust during all these years. I would also like to thank my pre-examiners, Professors Henk Wymeersch and Gonzalo Seco-Granados, for their invaluable input, which helped me to obtain new fresh perspectives and improve the quality of this study.

From the WISM2 project, I feel particularly grateful to Maurizio Bocca for his coaching, trust and assistance, even when he was already in Utah, to Ossi Kaltiokallio for preparing the software that controlled the wireless network during the data-collection experiments, and to Jari Saarinen, Mathew Myrsky and Antti

Maula for the existence of and assistance with J2B2, a crucial actor in this play. Finally, I cannot forget Jussi Timomen, a key person that made the project not only possible and successful but also exciting and really fun.

I would also like to thank in general my former colleagues from the *vanha labra* for their contributions to the friendly and pleasant atmosphere that still remains in the heads and harts of most (if not all) of us. Special thanks go to Panu Harmo, for all his support during my years in TKK and Aalto in all imaginable aspects. Since my arrival as an Erasmus student, he has been source of inspiration and wisdom beyond just university matters. Thanks also go to Jere Knuuttila, who was my first interface with Finland and introduced me to the sublime worlds of sauna, salmiakki and other aspects of Finnish culture. I would also like to thank José Luis Peralta for his complicity, close friendship and support throughout the graduation process, Janne Paanajärvi for his continuous predisposition to discuss scientific matters and offer his broad knowledge to all of us, and to Giusepe Peppe for all the world shared on two wheels.

From the Finnish Geospatial Research Institute, I would like to thank Heidi Kuusniemi and Rob Guinness for allowing me to use working time to finish this manuscript and for their patience waiting for the result.

And yes, there is life on the frontier of academic matters and behind it. I am grateful to have had many friends around that helped me digest better or else forget my long working days and refocus on other important things in life. For those remarkable lunch sessions, which fixed so many potentially lousy working days, I want to thank especially Jan Kallenbach, Tommi Kauppinen, Evgenia Litvinova, Andrè Schumacher, Carlos Lamuela and Mathew Turnquist, and for all the Otaniemi-free great life experiences, I want to thank especially Alberto García, Juan Gutiérrez, Javier Pérez, Fernando Losada, Manuel Jiménez, David Gambarte, Ruth Franco, Pablo Iglesias, María Razzauti, Sanna Koivu, Susanna Nenonen, Katja Nieminen and Anna-Kaisa Räisänen.

Finally, I should not forget my roots in Spain. Thanks to my friends from Villalba, with whom I can always 'go back to my old life' and without whom the escapes to my own country would not be so appealing and comforting. Special thanks to Guillermo and Enrique Alonso, Óscar Mulet and Daniel Rueda, who have been following my progress in every visit and with whom I have shared all the secrets of my life here *up north*. Also thanks to Fernando Díaz and Víctor Pérez, distinguished Erasmus co-adventurers in Finland, and to Jaime Tascón, who always enlivens the situation with his incomparable sense of humour. And finally, to my closest family: my sisters Sol and Zulema, my father, Cuca and Manolo, and last, but most importantly, to my mother, for her true and unconditional love and support.

Espoo, February 2016,
José María Vallet García

Contents

List of Figures

List of Tables

List of Symbols

Vectors and matrices are bold typed, and matrices are capitalized. Vectors with pure geometric meaning denoting position are represented with an arrow superscript.

Symbol	Meaning
$\backslash$	Backslash or left-divide operator: $\boldsymbol{A} \backslash \boldsymbol{b}$ is the vector $\boldsymbol{x}$ that solves $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$
$\equiv$	Equality denoting a definition
$\lvert \cdot \rvert$	Determinant, cardinality or absolute value
$\lvert\lvert \cdot \rvert\rvert$	Euclidean norm of vector $\boldsymbol{v}$
$\propto$	Proportional to
$\sim$	Distributed according to
$\mathbf{1}$	Column vector with all its elements equal to one (of full order)
$\mathbf{1}_{\text{L}}$	Column vector with all its elements equal to one (of reduced order)
$1_0(l)$	Indicator function
$\boldsymbol{A}$	Generic design matrix
$\boldsymbol{A}^t$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{\text{L}}$	Design matrix of the analytically linearized system of equations ($\boldsymbol{A}_{\text{L}}\vec{\boldsymbol{s}} = \boldsymbol{b}_{\text{L}}$)
$\boldsymbol{b}_{\text{L}}$	Measurement vector of the analytically linearized system of equations
$\text{Cov}\left[\mathbf{x}\right]$	Covariance of the random vector $\mathbf{x}$
d	Euclidean T-R distance
$(d\angle -\phi)$	Vector in polar notation, with magnitude d and angle ϕ
$\boldsymbol{D}$	Diagonal matrix of the K node-beacon distances (of dimension $K \times K$)
$\boldsymbol{D}_{\text{L}}$	Diagonal matrix of the $K-1$ node-beacon distances excluding the pivot (of dimension $(K-1) \times (K-1)$)
$\text{diag}(\boldsymbol{v})$	Diagonal matrix formed with the elements from vector $\boldsymbol{v}$
$\text{E}\left[\mathbf{x}\right]$	Expectation of the random vector $\mathbf{x}$
$h(l)$	Correlation function (of the lag l)
$\mathbf{I}$	Identity matrix
$\mathbf{J}$	Jacobian matrix
K	Number of beacons

$k_{\mathrm{c}}(l)$	Covariance function of the lag l
l	Lag, or distance between control points
l_0	Decorrelation distance
ln	Natural logarithm (base e)
log	Logarithm in base 10
n	Path loss exponent in the log-normal model
$p(z;\boldsymbol{\theta})$	Probability of z given the parameters $\boldsymbol{\theta}$
$p(z\|\boldsymbol{y})$	Probability of z conditioned on $\boldsymbol{y}$
$\vec{\boldsymbol{r}}_{\mathrm{k}}$	Position of the k-th beacon
r_{xk}	Coordinate x of the k-th beacon
r_{yn}	Coordinate y of the k-th beacon
$\boldsymbol{R}$	Robot's trajectory matrix $(\vec{\boldsymbol{r}}_1 \dots \vec{\boldsymbol{r}}_{\mathrm{K}})$
$\vec{\boldsymbol{s}}$	Position of the node
$\hat{\vec{\boldsymbol{s}}}$	Estimate of vector $\vec{\boldsymbol{s}}$
$\mathrm{Tr}(\boldsymbol{A})$	Trace of matrix $\boldsymbol{A}$
x	Coordinate x of the node
y	Coordinate y of the node
z	A generic observation from a model (RSS for the log-normal)
$\bar{z}$	Mean of the random variable z
z_0	Reference power in the log-normal model

Greek

γ	$\frac{\sigma \ln 10}{10n}$ ratio
γ_m	$m\gamma{=}\frac{m\sigma \ln 10}{10n}$, where m is an integer
Γ	$e^{\gamma^2/2}$
Γ_m	$e^{\gamma_m^2/2}$
ε	Random variable including IID and spatially correlated noise $(\zeta + \epsilon)$
ϵ	IID random noise
$\zeta(l)$	Spatially correlated noise modelling the (lag dependent) shadowing
$\boldsymbol{\theta}$	Generic vector of model parameters
$\boldsymbol{\theta}_{\mathrm{m}}$	Vector of parameters affecting the mean
$\boldsymbol{\theta}_{\mathrm{c}}$	Vector of parameters affecting the covariance matrix
ρ	Correlation coefficient
σ^2	Total variance
σ_n^2	IID noise variance
σ_c^2	Variance of the spatially correlated noise (shadow fading)
$\boldsymbol{\Sigma}$	Covariance of the noise vector measurements
$\boldsymbol{\Sigma}_{\mathbf{L}}$	Covariance of the transformed vector of measurements $(\mathrm{Cov}\,[\boldsymbol{b}_{\mathbf{L}}])$
$\boldsymbol{\Phi}$	Normalized covariance matrix (so that $\boldsymbol{\Sigma} = \sigma^2\boldsymbol{\Phi}$)

List of Acronyms

AGN	additive Gaussian noise
AoA	angle of arrival
AoI	area of interest
AP	access point
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BLUE	best linear unbiased estimator
BS	base station
CDF	cumulative distribution function
CEP	circular error probability
CM	common model
CRLB	Cramér Rao lower bound
DAoA	differential angle of arrival
DGPS	differential GPS
DRSS	differential received signal strength
EM	expectation maximization
ER	emergency and rescue
FIM	Fisher information matrix
FLOP	floating point operation
FLOPS	floating-point operations per second
GDoP	geometric dilution of precision
GFLOP	giga floating point operations
GLS	generalized least squares
GLS-ML	GLS using ML distance estimators
GLS-U	GLS using unbiased distance estimators
GLS-V	GLS using LMMSE distance estimators
GNLS	generalized non-linear least squares
GNSS	global navigation satellite system
GPR	Gaussian process regression
GPS	global positioning system

GS	grid search
GS-LBFGSB	L-BFGS-B initialized with a grid search
HDoP	horizontal dilution of precision
HW	hardware
IGLS	iterative generalized least squares
IGLS-ML	IGLS using ML distance estimators
IGLS-U	IGLS using unbiased distance estimators
IGLS-V	IGLS using LMMSE distance estimators
IID	independent and identically distributed
IM	individual model
L-BFGS	large-scale BFGS
L-BFGS-B	bound-constrained large-scale BFGS
LBS	location based service
LLS	linear least squares
LLSE	linear least squares estimation
LMIR	localization-model identification recursion
LMMSE	linear minimum mean square error
LoP	line of position
LoS	line of sight
LS	least squares
LSE	least squares error
MAP	maximum a posteriori
ML	maximum likelihood
MMSE	minimum mean square error
MoG	mixture of Gaussians
MSE	mean square error
NLS	non-linear least squares
OLS	ordinary least squares
PDF	probability distribution function
RMSE	root mean squared error
RSS	received signal strength
RTT	round trip time
SDP	semi-definite programming
SLAM	simultaneous localization and mapping
SW	software
T-R	transmitter-receiver
TDoA	time difference of arrival
ToA	time of arrival
USB	universal serial bus
VDoP	vertical dilution of precision
WLAN	wireless local area network

WLS	weighted least squares
WLSE	weighted least squares error
WSN	wireless sensor network

Glossary

Common model

A model used to predict the measurements associated with multiple nodes.

Control points

Positions associated with the measurements.

Control values

Measurements collected at the control points.

Fast fading

Same as small scale fading.

Geometrical configuration

Of a problem, it refers to the particular geographical disposition of the beacons and the node to be located.

Individual model

A model used to predict the measurements associated with a single node.

Lag

Distance between two control points.

Large-scale fading

Fading due to the presence of obstacles much larger in size than the carrier wavelength.

Pivot

Said of the beacon or its corresponding measurement equation chosen to be subtracted from the rest in order to form a linear system of equations in LLS position estimation.

Reference distance

Minimum distance for which an RSS-distance model is valid.

Reference model

Common model identified with data from all the nodes in one of our experiments in this thesis (one robot trajectory).

Reference power

Signal strength measured at the reference distance.

Shadow fading

Same as large-scale fading.

Shadowing

Same as large-scale fading.

Slow fading

Same as large-scale fading.

Small-scale fading

Signal strength variability in short distances due to multipath propagation.

Chapter 1

Introduction

Contents

1.1 Motivation and Objectives

Localization, within the present context, can be defined as a problem of position estimation. Humans and animals deal with this problem daily in a natural way, in the sense that most of the time we are not conscious of the problem that our brain is solving; it simply happens. We are travellers and wanderers, and therefore need to know where we are and how to get to different places. For that purpose nature and evolution have provided us with sophisticated sensing organs and a processing unit, the brain, capable of analyzing large amounts of information and yield results in real time.

Humans in particular have gone further. When our intelligence has allowed us to develop technology, we have been able to build artificial instruments and methods that enables us to estimate position and navigate in contexts for which our brain itself was, and is, not prepared. In that sense, these instruments complement our brain's capabilities allowing us to become aware of our position at higher levels of abstraction. Thanks to this ability, and following our internal exploratory instincts, we have been able to explore far lands, create routes for commerce and expand geographically our domains and civilizations.

As the technology has evolved, localization techniques have evolved with it, and this eventually made it possible to estimate position from anywhere on our planet. Industrialization and the development of automation and digital technologies have made further profound changes in the instruments used for localization purposes. We now have devices that can estimate position automatically, thus allowing end users to leverage the need for specialized orientation knowledge. In particular, outdoor localization has since been made available to the general public via global navigation satellite system (GNSS) technologies, especially that of global positioning system (GPS). This in itself has opened the door for more location based services (LBSs), further fostering the demand for more innovative localization technologies that could then open doors to new types of applications.

Context Awareness using WSNs: the Need for Localization

Currently, continuous evolution and miniaturization in microelectronics has made it possible to have computing power *everywhere and anytime*, a concept that is known as *ubiquitous computing*. Parallel to these advances, the development of radio-communications has made it possible to have generally available radios that can be used effectively for short-range communications in portable devices. The combination of ubiquitous computing and radio-communications together with sensing technology has given rise to the concept of wireless sensor networks (WSNs), which is a relatively new versatile and challenging practical paradigm for distributed applications.

WSNs are generally formed by small, low-cost and highly constrained devices

equipped with radios and diverse measurement sensors that can be deployed in large numbers. Once deployed, they can communicate with each other and send their sensory information to, e.g. a central server. From an engineering standpoint, the availability of large numbers of nodes provides robustness through redundancy to the overall system. Information-wise, it makes it possible to collect geographically distributed measurements, resulting in richer and more complete contextual information. This capability of acquiring more complete information makes WSNs a very attractive tool to be used in applications aimed at sensing environments, and more specifically in those targeted at building or enhancing context awareness.

The utility of WSNs in these types of applications relies completely on the ability to relate measurements to the position at which they are collected. The feasibility of this depends strongly on the type of application that the WSN is going to be used for. In the best-case scenario, the WSN can be deployed *a priori* placing the nodes in known positions within the desired environment. This implicitly means that a map of the environment is available whereby the nodes can be referenced. An example of this type of situation is the monitoring of an industrial process, in which the nodes are installed in well-planned, fixed positions. The floor-plan of the environment is unlikely to change, and if it does, the personnel can re-locate the nodes at will, always in known positions. Therefore, the association between information and its point of origin is guaranteed by design.

On the other hand, in the worst-case scenario the network will have to be deployed completely *ad-hoc*, placing the nodes at random unknown positions in an unknown and dynamic environment. Examples of such situations are emergency and rescue (ER) scenarios, such as fires, earthquakes and hostage situations. In this type of scenarios, when an emergency occurs a first responder has to travel to a specific place with all of their equipment within a short time, and typically with very limited or non-existing prior contextual information. The WSN will then be used to obtain and maintain context awareness, which will aid making decisions towards an effective resolution of the situation.

A concrete example of this type of scenario could be a building on fire. When the fire brigade first arrives to the scene, the information available to the team is very scarce and mostly consists of conjectures based on first arrival observations. It is clear that factual contextual information will be invaluable for tracing and maintaining an action plan. For that purpose, the fire fighters could deploy a WSN within the affected building to collect relevant information, such as the temperature in the rooms, the concentration of gases and the velocity of the wind, which will help understand what is happening inside the building. This information can be then used to, e.g. monitor the structural status of the building, predict the evolution of the fire and gases and select accordingly in real time the best actual and future evacuation routes.

The deployment of the network can be done in different ways. However, given the chaotic nature of ER situations and the typical lack of floor plans or maps, the resulting position of the nodes is usually unknown independently of the deployment method. Some of the nodes might be naturally mobile, for example those attached physically to the rescue team members. Additionally, those placed in fixed positions might move, perhaps because they simply drop, are accidentally kicked or because of the collapse of parts of the structure. Therefore, in the most general case, using WSN for context awareness in ER scenarios requires the capability to automatically localize the nodes that form the network.

Robot-Assisted Localization

Radio localization methods typically require the availability of some devices with known location that can transmit or receive radio signals. These devices can then be used as references with the help of which the rest of the nodes can be located. However, in ER situations one cannot realistically assume the availability of a pre-installed infrastructure in the area of interest. As already argued, in the most general case the first responder team will have to install the required equipment as they arrive, including the WSN. Therefore, in the most generic scenario, the nature of ER situations requires localization methods that do not rely on an existing pre-installed infrastructure.

In this thesis, we propose and investigate the possibility of using a robot as a mobile beacon for localization in WSNs. Mobile robots can be equipped with complex and expensive sensors with which they can map the environment and estimate their position within that map. Because their position is known at all times, they can act as mobile references for the rest of the nodes, and therefore leverage the need for a pre-installed infrastructure. In particular we will use a single robot to locate the nodes of a WSN using received signal strength (RSS) measurements and model-based algorithms, although the methodology can also be exploited when using more robots and other types of observations (e.g. time of arrival (ToA)).

Using mobile robots as a localization assistive devices has other advantages besides eliminating the need for a fixed infrastructure. As mentioned above, they can generate a map of the environment. This map can be used as a global reference for any other location-based service required. This eliminates the problem of establishing a coordinate system, which is a fundamental step in relative localization. Additionally, being mobile, the robots can collect large numbers of observations in more geographically distributed areas as opposed to using fixed sensors. This results in richer datasets containing more information, which can be exploited to produce better position estimates in two ways: first, intuitively, localization algorithms are expected to produce better estimates when more information is available; second, the fact that the measurements are geographically

distributed and available in large numbers makes it possible to characterize the radio-propagation environment locally. This geographical specialization of the model results in better predictions of the RSS in all the environment, which in turn can increase significantly the performance of localization algorithms.

Scope and Objectives of the Thesis

Despite the great potential advantages that using mobile robots could offer in many practical situations requiring localization services, the reality is that they are seldom used in real scenarios for that purpose. The work presented in this thesis is a small step towards understanding, leveraging and fully exploiting the capabilities of mobile robots as localization-assistive devices in many applications.

The main research questions addressed are: a) How can we exploit the rich spatial information provided by a mobile robot to estimate the position of the nodes of a WSN as accurately and precisely as possible? b) Furthermore, how can we do it in a computationally efficient manner? c) And last but not least, how can this be done in real *ad-hoc* deployments with no *a priori* information about the environment?

The work presented here is grounded in RSS-based position estimation using parametric models, and as such the most general goal is to extend the state of the art by proposing new practical and computationally efficient localization methods. In order to address the first two research questions, we conduct studies pertaining RSS modelling and algorithms development. The third question is then addressed by proposing localization algorithms that do not require model identification prior to the position estimation.

The means by which the WSN is deployed is not within the scope of this thesis, nor is the technology required in order for the robot to be able to map the surroundings and locate itself. That is to say, we depart from the assumption that the nodes have been deployed by *some means*, and that a simultaneous localization and mapping (SLAM) capable robot is readily available for our purposes.

The steering agent is the applicability in context awareness, particularly in ER scenarios. Therefore, we put a strong emphasis on the practicality of the solutions proposed. For that purpose, the algorithms introduced in this study have been tested with real data gathered in three different environments, each with distinct characteristics (see Section 1.4). However, we recognize the large difference between a real ER scenario and the conditions under which the data have been collected in our laboratory tests. Hence we do not consider that the methods proposed are ready to be used in all possible types of real emergencies scenarios, but rather that we have made a small contribution towards that possibility.

1.2 Localization Methods

The problem of estimating the position from which a signal emanates is known in scientific literature as the *source localization* problem. In this thesis the aim is to locate the nodes of a WSN using a robot that sends radio packets. Therefore the source of the signals is the robot, and not the nodes that we want to locate. However, by the principle of reciprocity the situation is equivalent to the one in which the node sends and the robot listens (see Section 1.3). Therefore, source localization algorithms can be directly applied in our problem, and in fact they are a central topic in this work.

There are several radio-signal measurements and positioning methods that can be exploited for source localization. Regarding the type of signals observed, we primarily have signal strength, propagation time and angle of arrival (AoA). RSS-based methods use the signal strength of arriving packets as the main observation from which position will be inferred. Time-based approaches instead use the time that it takes for a signal to travel from the transmitter to the receiver (ToA), the round trip time (RTT) or the time difference of arrival (TDoA). Finally, AoA exploits measurements of the relative angle between the radios. Our interest is mainly in RSS, simply because the radios that we have available already provide that particular signal. This is a direct consequence of the low complexity and price of the circuitry necessary to measure signal strength, which contrasts with the intricacy of the technology required for measuring propagation time. Although time-based technology has a great potential for localization [Gezici et al., 2005, Gustafsson and Gunnarsson, 2005], the reality is that there are few commercial, off-the-shelf radios currently available with which time-related measurements are possible, whereas RSS is available in almost all radios.

With respect to the localization algorithms, different classifications are possible. For example, Seco *et al.* distinguish four main groups attending to the mathematical method used: geometry-based methods, minimization of a cost function, fingerprinting and Bayesian techniques [Seco et al., 2009]. Agrawal and Patwari consider three main categories: model-based, kernel-based and fingerprinting algorithms [Agrawal and Patwari, 2011]. We consider here a broad categorization into two major groups attending to the degree of dependency on training data prior to the exploitation or production phase: model-based and pattern matching, also known as fingerprinting.

Model-based methods aim at finding simple parametric statistical models that relate the RSS to the transmitter-receiver (T-R) distance. The models are then exploited in different ways for position estimation. Perhaps the most widely used RSS-distance model in the scientific literature is the site-specific log-normal model proposed by Seidel and Rappaport in [Seidel and Rappaport, 1992]. It consists of a straight line decaying with the log-distance plus a normally distributed error,

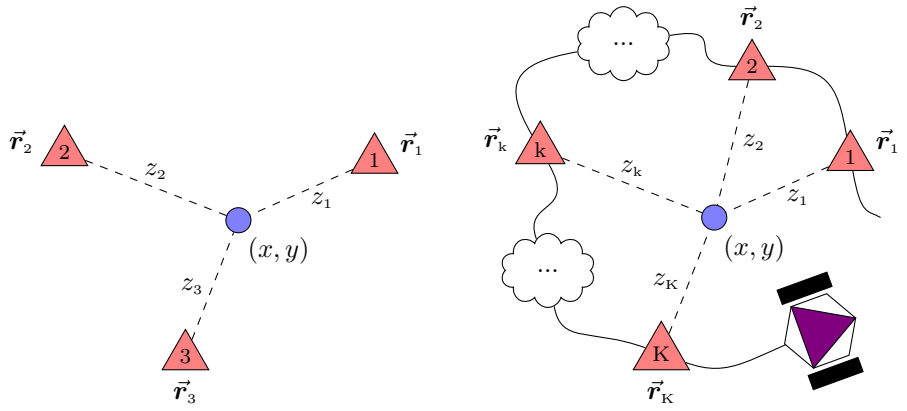
and is characterized by three parameters that depend on the hardware (HW) used and the aggregate characteristics of the radio environment (see Section 2.5). For an optimal performance of the localization methods, these parameters must be estimated using training data collected in the specific environment where the model is to be used. After identifying the model, the training data is no longer necessary for estimating position.

One of the main challenges of RSS-distance model-based localization methods is that simple parametric models work well only in very homogeneous environments, and they are not good predictors in the typically complex indoor settings. Acknowledging this difficulty, fingerprinting localization methods do not aim at finding a formal RSS-distance dependence. Instead, their goal is to produce a direct mapping between the RSS and the position. They rely completely on a training phase in which vectors of RSS measurements associated with fixed reference devices are stored together with the position in which they are collected. Position estimates are then produced directly using a sub-set of the training data that *best matches* the observations made during the production phase, and therefore the entire training set is required for each position estimate. The advantage of fingerprinting methods relies in the better RSS predictions produced by the direct RSS-position mapping compared to predictions produced by the over-simplistic parametric models. The disadvantages are that they cannot produce position estimates in areas where no training data have been collected, and the substantial demand for computing power when the database becomes large.

There are methods that use the best of both worlds. For example, Gaussian process regression (GPR) can use parametric models to predict the overall mean, while adjusting the predictions to the data observed during the training phase [Ferris et al., 2006, Muppirisetty et al., 2015]. Other kernel methods have also been used for localization purposes, in which a model is not used to directly predict the RSS, but rather to evaluate the degree of similarity between samples based on which the position can be estimated [Agrawal and Patwari, 2011].

In many practical situations, it is not realistic to assume the availability of training data for the area of interest. Still, should one have proper training data, the environmental conditions might change significantly, thereby reducing the utility of the possibly available training data. ER situations are perhaps an extreme example, in which the environment is most likely completely unknown and highly dynamic. These scenarios pose a restriction on the type of localization algorithms that can be used. It is desirable to use methods that do not rely on prior training data and that are capable of adapting to changing conditions in the environment. These requirements rule out methods that depend unconditionally on training data, such as fingerprinting.

Model-based methods also rely on training data, but only indirectly through the model parameters, which is a much less restrictive type of dependency. The



(a) Localization using 3 beacons.

(b) Localization using a mobile robot.

Figure 1.1: Equivalence of localization using fixed beacons and a mobile robot.

lack of information on some of the parameters in a statistical model is a well-known problem for which several solutions are available. One can, for example, estimate the model parameters together with the position of the target node, or else simply treat them as nuisance quantities that can be marginalized out. This makes them suitable for localization in dynamic environments and with no prior training data. For these reasons, our work is based on parametric RSS-distance models. In particular, our work will be based on linear least squares (LLS), maximum likelihood (ML) and maximum a posteriori (MAP) estimation.

1.3 Localization Using a Mobile Robot

The problem of locating the nodes of a wireless network can be addressed from different perspectives. Quite often it is assumed that the position of some of the nodes is known *a priori*. These nodes can then be used as beacons or references to localize the other nodes. Panel 1.1a depicts this approach when using 3 reference devices. The red triangles represent beacons, the blue circle a target node to be located and z_k the observations based on which its location is to be inferred (e.g. ToA or RSS).

In this thesis we take a different approach: we assume that the position of all the nodes is unknown, and we use a SLAM-capable robot as a mobile beacon. While the robot moves around, it builds a map of the environment using its laser scanner and odometry sensors. Thus, its position within the map is known at any moment. At the same time it is communicating with the rest of the nodes of

the network. The fixed nodes can measure the RSS of packets coming from the robot. Because the position of the robot is known at those particular instants, each of the RSS measurements can be thought of as coming from a beacon placed at the position of the robot at that particular instant. In this sense, the robot is acting as a mobile beacon, and the node localization problem can be formulated equivalently as if we had K beacons in fixed positions, with the particularity that each of the beacons transmits only one packet. Panel 1.1b depicts this concept. The black curve represents the trajectory followed by the robot. At the k-th measurement instant ($k = \{1, \ldots, K\}$), the robot is at position $\vec{\boldsymbol{r}}_{\mathrm{k}}$ and the node measures z_{k}. The problem is mathematically equivalent to one in which beacons (red triangles) are located at positions $\vec{\boldsymbol{r}}_{\mathrm{k}}$.

The concept of using robots to locate WSN nodes has already been studied in the existing literature. One of the first attempts can be found in [Sichitiu and Ramadurai, 2004]. The authors use a remotely controlled truck equipped with a GPS to move a hand-held device outdoors that broadcasts messages containing information about the location of the truck. Nodes receiving these messages construct '*probabilistic constraints*' based on the location of the truck and distance estimates from RSS. The reported localization error was less than three metres.

In [Pathirana et al., 2005], a mobile robot is used to estimate the location of the nodes of a network organized in clusters in a large indoor area. The authors used a logarithmic RSS-distance model with a random variable with unknown probability distribution function (PDF). They also identify the model *a priori* before the experiments. Because the PDF of the noise was considered unknown, a robust extended Kalman filter was used to estimate the position of the nodes. The reported localization accuracy is approximately one metre.

In [Caballero et al., 2008a], a robot equipped with differential GPS (DGPS) moved around an outdoor parking lot measuring RSS from neighbouring nodes. The localization algorithm at first applies a particle filter for an initial approximation of the nodes' position using measurements from the robot, and then an information filter for refining the estimates using measurements among the nodes. The likelihood of the measurements is modelled as a Gaussian whose mean and standard deviation increase with the distance. The model was identified *a priori* before the experiments and, to account for the differences among the nodes, the authors intentionally inflated the standard deviation. A similar particle filter and model were used in [Caballero et al., 2008b], but this time a helicopter was used as a mobile beacon to estimate the position of the nodes in 3D.

In [Menegatti et al., 2009], an extended Kalman filter was used to simultaneously estimate the position of a robot and map the nodes using distance estimates from RSS and odometry from the robot. The authors used a log-normal RSS-distance model, which was first identified before conducting the experiments and then used to calculate the inverse distance-RSS model. The tests were car-

ried out indoors in a large corridor. The reported accuracy is less than a metre when no prior information about the location of the nodes is available and less than half metre with priors close to the true positions.

In [Caballero et al., 2010] the authors complement their RSS measurements model from [Caballero et al., 2008b] with a mixture of Gaussians (MoG) representing a multiple hypothesis over the angle of arrival of the signal. Upon the arrival of the first RSS measurement, the Gaussian components are equally spaced throughout the whole angle. As the robot moves, new RSS measurements are integrated using a Kalman filter. The parameters and mixture coefficients of the MoG are then updated based on to the likelihood of the new measurements, and Gaussians with poor weights are pruned. This procedure allows to jointly estimate the position of the robot and all nodes involved.

All of the approaches reviewed so far have several characteristics in common: a) they use geographically sparse measurements, thereby not fully exploiting the potential of having a robot available that could generate a high rate of samples, b) all the experimental tests are done in similar environments, c) they all use one RSS-distance model common to all the nodes and d) the parameters of the model are estimated *a priori* from training data before the real experiments take place.

In this thesis, we present methods that differ in all these respects from previous studies in the existing literature, with the ultimate goal of offering better localization accuracy. First, we test our algorithms in three different environments with clearly distinct characteristics. Second, we sample the RSS with a higher rate in order to obtain more spatial information, resulting in a dense line of measurements associated with the robot's trajectory. Third, we explore the possibility of using one RSS-distance mode per node instead of a global one for all the nodes. This helps mitigate the effects of the HW variability characteristic of WSNs and, to some extent, the shadow fading. Finally, at the end of the thesis we will present localization methods that do not rely on prior deterministic estimates of the model parameters, making them more suitable for practical ad-hoc deployments characterized by a lack of prior knowledge.

1.4 Experimental Setting

In this section we explain in detail the experimental setup used to gather the data necessary to carry out the studies included in this thesis.

The base working scenario is one in which a WSN is deployed in an area of interest to collect some data for purposes that are not relevant to this study. The nodes are assumed to be placed in random positions in the sense that the placement is not necessarily planned in advance. The method of deployment is not considered in this work, although one can imagine e.g. a person exploring the area of interest and randomly placing the nodes throughout the exploration.

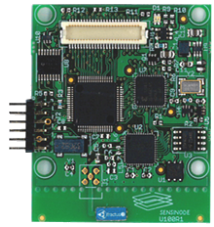


Figure 1.2: Micro node platform.

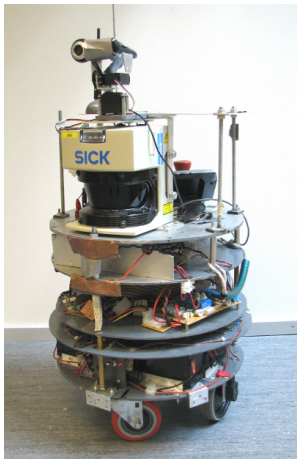
A mobile robot is commanded to explore the area following random trajectories. The robot is part of the network and thus is naturally communicating with the rest of the nodes. While this communication is occurring, both the robot and nodes measure the RSS of all packets sent by any transmitter from the network. The robot is able to estimate its own position at all times using its laser scanner and odometry sensors. This means that any RSS measurement can be associated with the position of the robot at any instant. The dataset collected from this setup consists of the RSS measurements and their associated robot position[1]. In the remainder of this section we explain the relevant details of the infrastructure and procedures pertaining the collection of the data.

The WSN

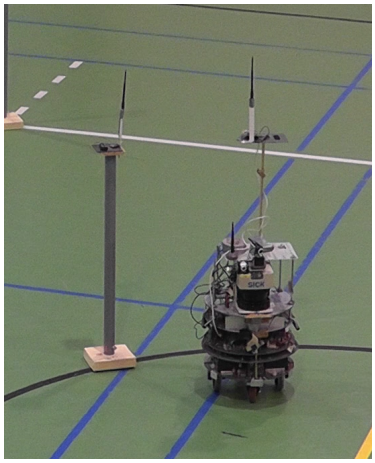
The WSN was formed by nodes from the micro series from the company Sensinode (see Figure 1.2). The core of the nodes is a MSP430 low-power microcontroller and a TI CC2420 802.15.4 compliant radio using the 2.4 GHz ISM band. All of the nodes were equipped with commercially available omnidirectional antennas.

The communication protocol among the nodes followed a token-passing scheme using a predefined set of frequencies. During each round of communications, each node in sequential order broadcasts a packet to all the others using the same frequency. Each node has a unique ID that is used to uniquely identify the origin of the packets. The listening nodes then measure the RSS of the packet sent by the transmitter in turn. At the end of a complete round, all the nodes have the RSS measurements of the packets sent by their neighbours in that communication round. This information is stored and sent within the packet which the nodes will send during the following round of communications. After every round of communications, the nodes synchronously switched to the next radio channel in the schedule. The channels chosen were those that showed less noise prior to the experiments.

[1]From this dataset, only the RSS from packets transmitted by the robot and measured by the nodes will be used for localization in this thesis.



(a) J2B2, the robot.



(b) J2B2 in action.

Using the previous simple communication protocol, any node had access to all information sent by the other nodes. One of the nodes was mounted on the mobile robot and connected to its computer via a universal serial bus (USB) cable. This node, besides taking part in the ordinary communication cycle of the WSN, forwarded all of the RSS measurements from all the nodes to the robot via the USB cable during every round of communications. The data was then stored in the robot for off-line analysis. The available data contain the RSS measurements for each pair of nodes and each communication cycle.

The Robot

The robot used was a typical differential-driven research platform. We call it J2B2 for short (see Panel 1.3a). J2B2 is equipped with a laser scanner and encoders, and it is capable of simultaneous localization and mapping (SLAM). The communication with the control station is done using standard 802.11b wireless local area network (WLAN). The core of the robot consists of a Linux PC running GIMnet, a publisher-subscriber communication infrastructure designed for robotic applications. It enables HW and software (SW) abstraction, modularity and reusability. Thus, it makes it possible to use the same HW, control modules and user interfaces in different robots with minimal configuration changes [Saarinen et al., 2007].

One of the utilities developed as part of the GIMnet software package is a data logger, with which all the data running through the robot during an experiment can be timestamped and stored in the robot itself or in a remote computer. In

addition to the data from the encoders, laser scanner, camera and other internal equipment of the robot, the data coming from the WSN through the USB port is also timestamped and stored. In this way, the RSS measurements can be associated with the position of the robot at any time.

The Environments

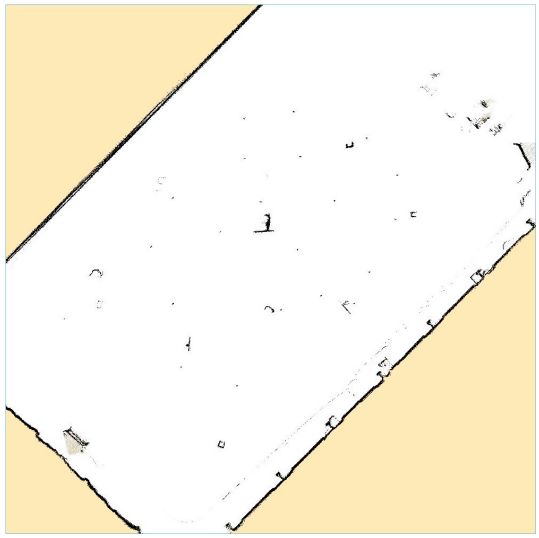
One of the goals of this thesis is to explore methods that can be applied in different scenarios. Thus, it was considered necessary to gather data in environments with different propagation characteristics, which can the be taken as representative of a broad range of scenarios. One rough classification of the various types of environments is indoors versus outdoors. This classification is quite loose, as the differences between environments within any of these categories can be very large. Another classification could be done attending to the degree of multi-path propagation present in the environment. Considering the limitations we had with respect the environments available for gathering data[2], we selected three that were deemed representative of typical scenarios:

- A large basketball field (see Figure 1.5a). This scenario is representative of environments with a continuous clear line of sight (LoS). As we can see in the figure, the field is almost empty with no significant obstacles. However, some amount of reflection occur, mainly from the walls, floor and ceiling.
- The lobby of a building (see Figure 1.5b). This scenario consists of large open spaces, but contains more significant obstacles than in the basketball field, some of which are large and dominant (e.g. the stairs, large plant pots and a large metallic door). Thus the environment is partially obstructed and has less clear LoS. Many more reflections are possible with respect the situation in the basketball field.
- A typical modern office (see Figure 1.5c). In this scenario there are more reflecting surfaces and corners where the signals can be diffracted, and the propagation is more difficult in general.

The experiments

For the experiments we deployed 18 to 20 nodes in random fixed positions in the three environments, which covered an approximate area of 20x20, 20x30 and 12x20 m^2 for the basketball field, lobby and office respectively. The nodes were mounted on roughly 1 m high poles. Considering a transmitting distance of 20 m and a 2.4 GHz frequency of the carrier, the maximum radius of the first Fresnell

[2]The experiments were carried out in Finland in winter time. The experiments could not be made outside, as the snow hindered the movement of the robot and compromised its integrity.



(a) Basketball field

(b) Lobby

(c) Office

Figure 1.4: Matched maps for the different environments used in the experiments. The maps are not in the same scale.

(a) Basketball field

(b) Lobby

(c) Office

Figure 1.5: Pictures of the different environments used in the experiments.

zone is approximately 0.8 m. Thus, the poles ensure that the floor does not obstruct the first Fresnell zone.

Three independent tests were carried out in each of the environments. For each test the robot followed a random trajectory as it tried to cover the whole area of interest. Each test lasted approximately seven minutes, during which the WSN operated as described before with a communication rate of approximately 10 rounds per second and using eleven channels. The data collected were stored in the robot for later offline processing.

All of the experiments were made trying to minimize the effects of humans moving around. In this sense, we can consider that the environments were nearly static (obstacle-wise) during the duration of the experiments.

The RSS signals used for node localization consisted of packets sent by the robot and measured by the nodes themselves. The signals were averaged over a distance of 25 cm travelled by the robot and over all the communication channels. This reduced the effect of fast fading and standardized the density of effective measurements along the robot trajectory.

The maps from different tests for the same environment were matched as described in [Saarinen et al., 2011] to obtain a unique reference map and global coordinate system for each environment (see Figure 1.4). These constitute the frames with respect to which the nodes' position estimates will be referenced.

The Nodes' Ground Truth Position

In order to asses the performance of the proposed localization algorithms, we need to compare the estimates with the true positions. Therefore, we need to measure precisely the position of all the nodes in our deployments.

Measuring the position of randomly placed nodes in different environments can be done in different ways. Often it is done following manual procedures, e.g. using measuring tape and perhaps relying on the squareness of the corners of buildings. These methods take time and require careful and precise manual measurements, and therefore they are error prone.

In this study, in order to obtain the ground true position of the nodes we take advantage of the natural capability of the robot to detect obstacles. Towards this end, the poles that the nodes are mounted on top of are thick enough that they can be seen by the robot's laser scanner (see panels 1.5a and 1.5b of Figure 1.5). Thus, equally as any other static obstacle, they appear on the maps shown in Figure 1.4, from where their position can be measured. Their small cylindrical shape causes them to appear as tiny dots, which helps to estimate their position accurately. The recognition of the poles in the map is done manually by a human aided by a sketch of the position drawn after the nodes have been deployed. Their position is then calculated based on the coordinates of the dots in pixels and the resolution of the map (in metres/pixel). These coordinates served as the ground

true position of the nodes, with an estimated error in the order of centimetres. This error is assumed to be negligible with respect to those achieved when using typical RSS-based localization methods.

1.5 Thesis Outline

Chapter 2 presents a perspective on existing RSS-distance models. Having a good model[3] is critical to the performance of model-based position estimators. In this thesis, the classic log-normal model will play a fundamental role, and therefore we consider that it is important to reflect on its origins as well as on its advantages and limitations when used in real WSNs deployments, especially indoors. Reflecting on these, and particularly on the limitations, is the first step in finding new ideas to overcome the shortcomings. In particular, we elaborate on how the scale of the network and the homogeneity of the environment impact its usability given the radial nature of the model. This is especially important in WSN, where typically the communication ranges are relatively small with respect to the size of the obstacles in the environment.

Chapter 3 is dedicated entirely to an empirical characterization of the log-normal model using the data collected from our experiments. It is well known that its parameters depend on the experimental conditions. Traditionally, site-specific models are identified just once for a given environment using a single T-R pair of devices. This model is then used for the localization of any device within the site, a methodology that we refer to as using a common model (CM) for localization. Using CMs for localization purposes in WSNs gives rise to questions regarding the inevitable HW variability observed in low-cost nodes and the local variability of inhomogeneous environments. How much do these factors affect the model parameters and what is the impact on the performance of localization algorithms? Is it something that we can really ignore? In Chapter 3 we present a detailed study of the parameters' variability together with a comprehensive explanation of the factors that affect them, all grounded in experimental observations. An analysis of the impact of this variability is left to Chapter 5, after discussing in Chapter 4 the core localization techniques that will be used during the rest of the thesis.

Chapter 4 presents a detailed study of several existing and newly proposed source localization methods based on the least squares error (LSE) criterion. Throughout the chapter we consider independent and identically distributed (IID) measurements, a convenient common mathematical assumption. Results are presented for the log-normal and additive Gaussian noise (AGN) models. We review and compare via simulations two different problem propositions that

[3] In the sense that it predicts as faithfully as possible real RSS measurements.

lead to their respective non-linear least squares (NLS) problems (including ML). Later we revisit a classical analytical linearization technique that results in a LLS problem that can be solved using ordinary least squares (OLS). Despite its advantages the technique is not used much in practice because of its relative large inaccuracy, especially in RSS-based localization. We then extend the range of existing linear localization methods by proposing the families of generalized least squares (GLS) and iterative generalized least squares (IGLS) algorithms inspired by classic estimation theory. After comparing all the methods, we conclude that ML is still the best in terms of general accuracy. However, the new linear methods proposed here show a significant performance improvement with respect to OLS, rendering them as a realistic alternative to ML localization especially in computational constrained applications.

Chapter 5 is dedicated to two main inter-related topics. First, to validate experimentally the source localization methods studied in Chapter 4 when using a robot as a mobile beacon; and second, to investigate the effects of the model parameters' variability observed in Chapter 3 on ML localization.

We start by studying how using a high linear density of RSS samples affects the log-likelihood function, an issue that arises naturally when using a robot as a mobile beacon. In essence this can create local maxima with a predictable characteristic geometry in which the solvers can get trapped. We then propose a new method, which we call Maximize-Explore, to overcome these specific types of local maxima. We continue with an experimental comparison of the performance of the best source localization methods selected from Chapter 4, in which we emphasize the contrast between using common and individual models. Using individual models (IMs) allows the models to capture individual HW and local environmental characteristics, which then can cope better with the variability in the model parameters observed in Chapter 3 than can a single common model. This model specialization can lead to significant improvements in the performance of ML localization. Motivated by the results, we continue with a study showing the sensitivity of ML localization to variations in the model parameters, and elaborate further on the use of common and individual models. Finally, we study the possibility of using polynomial models of a higher degree as well as MoGs to further specialize the IMs.

Although quite convenient in practice, the IID measurements assumption used so far is not realistic. It is well known that shadow fading induces correlation among RSS samples, and this becomes especially relevant when using a high spatial density of measurements. Chapter 6 is dedicated to localization methods that specifically take into account the spatial autocorrelation among observations. We first choose a theoretical model for the autocorrelation; we then use the model to demonstrate analytically that ignoring it can lead to overconfident position estimates, a problem that we identified back in Chapter 5. We invest-

igate further two possible solutions using our experimental data: a) discarding correlated measurements and b) incorporating the autocorrelation into the measurement model. When using ML position estimation, the first solution results in a gradual increase of the mean error as the number of discarded samples increases, whereas the second option automatically corrects the over-confidence problem while maintaining a low mean error.

Until this point in the thesis, all of the proposed localization methods rely on identifying the RSS models *a priori*. Our last point of focus, presented in Chapter 7, is devoted to the problem of position estimation when the model parameters are not known *a priori*. We develop and evaluate three different methods based on the joint estimation of position and model parameters and validate them using the data from our experiments. Two of the methods rely on the use of concentrated log-likelihoods, a technique that reduces the dimensionality of the input space and results in more efficient algorithms. In all cases, the estimators are shown experimentally to be quite sensitive to the geometrical configuration and the initial conditions given to the solver. We then propose a technique that solves the problem by indirectly applying regularization on the parameters affecting the mean of the predictions, and demonstrate its effectiveness in all of our environments.

Finally, Chapter 8 recapitulates the work in its entirety presenting a summary of the main conclusions.

1.6 Scientific Contributions

The main contributions of this thesis are localization algorithms and the studies that lead to and validated them. These methods have been tailored to the problem of using a robot as a mobile beacon for node localization in WSNs, in which special attention has been paid to the particular challenges that this paradigm poses. The line of research has led to algorithms that can be used without prior model identification, with a demonstrated sub-metre accuracy in our three experimental laboratory setups. This result opens up the applicability of the proposed methods to many practical applications in which WSNs need to be deployed *ad-hoc*.

The remainder of this section presents a list of the specific contributions.

From Chapter 3:

- The chapter assesses the variability in the model parameters when using measurements from nodes of a WSN for model identification in different types of environments. The work reveals explicitly their variability due to HW and local environmental particularities. The study includes an analysis

of the impact of the antenna directivity on the model parameters and the cause of correlation between them (sections 3.2 to 3.6).

From Chapter 4:

- Chapter 4 as a whole presents a systematic, comprehensive and self-contained study of source localization algorithms grounded on least squares (LS) and ML optimization criteria under the log-normal and additive Gaussian noise models. The study presents methods based on classical estimation, recent research on distance-based algorithms and newly proposed linear position estimators (see below) in a common coherent framework including a theoretical analysis and simulation-based comparisons. We believe that the text can be used as a guide for new students in the field as well as reference material for experienced researchers.

- The novel GLS and IGLS families of linear localization methods under the log-normal and additive Gaussian noise models, including an efficient and numerically robust implementation (Subsection 4.5.2).

From Chapter 5:

- Two problematic side-effects of using a high spatial density of RSS measurements on the log-likelihood function are identified, namely the appearance of a characteristic type of local maxima and the over-confidence of ML position estimators under the IID assumption (Section 5.1).

- The Maximize-Explore algorithm proposed as a means of overcoming the local maxima characteristic of this problem (Section 5.2).

- A study on the sensitivity of ML position estimation to variations in the log-normal model parameters. This work exposes the relevance of the variability on the model parameters presented in Chapter 3 (Section 5.4).

- A quantitative study of the effect of using data from many nodes for model identification on the performance of ML localization (Subsection 5.5.1).

- A quantitative comparison of using IMs and CMs in ML localization (Subsection 5.5.2).

- A study on the possibility of using models with generic polynomials and mixture of Gaussians in ML localization compared to using the classical log-normal model (Section 5.6).

From Chapter 6:

- A study on the effect of discarding correlated samples on ML localization (Section 6.5).
- The proposition (Subsection 6.1.2) and empirical characterization (Section 6.4) of a model for the spatial autocorrelation of RSS samples.
- The theoretical demonstration that, under the proposed model, ignoring the spatial autocorrelation can, but does not necessarily, result in over-confident position estimates (Section 6.3).
- Proposition and validation of using ML localization under the log-normal model including spatial autocorrelation to estimate the position of the nodes solving the over-confidence problem (Section 6.6).
- The mathematical formulation of the classical OLS as well as the new GLS and IGLS linear localization methods including the effects of exponentially decaying autocorrelation (Section 6.7).

From Chapter 7:

- The localization-model identification recursion (LMIR) iterative method to jointly estimate position and model parameters and its application using the polynomial models of Section 5.6 (Section 7.3).
- The formulation and usage of the concentrated log-likelihood function (including spatially autocorrelated measurements) with respect to some of the model parameters ($\boldsymbol{\theta}_{\mathrm{m}}$ and σ) for position estimation (Section 7.4).
- Diagnosis of one type of identifiability issue associated with using concentrated log-likelihoods for localization under the log-normal model, together with an experimental study that reveals how to detect it (Subsection 7.4.2).
- The proposition and demonstration of the effectiveness of a MAP based technique for solving the identifiability issues detected using regularization while at the same time conserving the low-dimensionality of the concentrated log-likelihood input space (Subsection 7.4.2).
- The formulation and usage of the concentrated log-likelihood function (including spatially autocorrelated measurements) with respect to the position. This approach is possible thanks to the good performance of the GLS methods developed in the previous chapters (Section 7.5).

The author of this thesis deems that the most important contributions are a) the two new families of linear position estimators from Subsection 4.5.2 (GLS and IGLS), due to their excellent accuracy to computational complexity ratio and the

elimination of the need for an initial position estimate; b) the consideration of and treatment given to the spatial autocorrelation among measurements, a matter typically ignored in the localization literature; and c), the MAP based algorithm for joint localization and model identification presented in Section 7.4.2, which takes into account the spatial autocorrelation among observations and enables position estimation in unexplored and dynamic environments taking advantage of the high-rate sampling capabilities of, e.g. mobile robots.

1.7 Author's Contributions

The work presented in this thesis was carried out within the framework of the WISM2 project (Wireless Indoor Situation Modeling), from which the author used the existing infrastructure to collect the necessary experimental data. In this sense, elements of particular importance were the WSN and the mobile robot.

The infrastructure related to the WSN was prepared by Ossi Kaltiokallio and Maurizzio Bocca, including the communications protocol stack, general SW configuration, HW-related issues and the practical arrangements required to carry out the experiments.

The robot (J2B2) was designed, built and programmed by Jari Saarinen and his team through years of research. Besides Jari himself, Antti Maula and Mathew Myrsky prepared the robot for the experiments and assisted with the data collection. These are also the key persons responsible for the SW that governs the robot. Finally, all of the SLAM and map-matching algorithms used to locate the robot are the result of Jari's research and his team, which the author of this thesis used conveniently for the purpose of his research. The author designed the experiments of the measurement collection campaigns, coordinated the efforts of both the WSN and robot teams and conducted the field tests with them.

The author's scientific contributions start after the assumption that the robot's position is perfectly known at all times, and hence can be used as a mobile beacon. All of the contributions described in Section 1.6 are therefore solely the work of the author. Some of the results from this thesis have been published in [Vallet et al., 2012], [Vallet et al., 2013] and [Vallet García, 2015], and the outcome of the WISM2 project in [Bjorkbom et al., 2013]. While preparing the first two papers, the author benefited from lively discussions with and feedback from his colleagues that helped him to prepare, shape and write the publications, and which stimulated later studies performed by the author.

All of the required programs and algorithms have been prepared by the author using Octave [Eaton et al., 2014] (Matlab compatible) and C/C++. During his journey, the author has also used code from many open source projects and contributed back to some of them with improvements and bug fixes.

Chapter 2

A Perspective on RSS-Distance Models

Contents

The fundamental measurement that we will use to estimate the position of the nodes is the RSS, that is, the power measured by a receiver. The medium in which the wave propagates from the transmitter to the receiver constitutes the communication channel, and its characteristics will have a decisive impact on the RSS. It is then of crucial importance to understand how the communication channel affects the RSS.

The aggregate effect of the channel on the signal strength is usually described in terms of *path loss*, which is a positive quantity that represents the attenuation in dB of the signal when it traverses from the transmitter to the receiver. It is defined as

$$PL[\text{dB}] = 10 \log \frac{P_t}{P_r}, \tag{2.1}$$

where P_t and P_r are the transmitted and received power, respectively. Once we know the path loss and the transmitted power (e.g. in dBm), the received power can be calculated as

$$P_r[\text{dBm}] = P_t[\text{dBm}] - PL[\text{dB}]. \tag{2.2}$$

As the path loss is a positive quantity, the received is less than the transmitted power.

This chapter presents an overview of some of the existing RSS models, while keeping an eye on how they have been appeared as radio communications and cellphone technology have continued to evolve. Emphasis is placed on the need for empirical models due to the complexity of the multi-path radio-propagation environment, and the fact that a gradually reducing the scale of the networks required new models that better explain the phenomena predominant in the different scales. The goal is to eventually present the log-normal model from its origins within the context in which it was originally proposed in classical studies. This will help us reflect on its virtues and limitations, and especially on its usability in WSNs.

The material presented in this chapter is by no means a complete and/or detailed description of all the available models, and interested readers are invited to consult classical textbooks, such as [Bertoni, 1999], [Parsons, 2000], [Rappaport, 2002] and [Goldsmith, 2005].

2.1 From Theoretical to Empirical Models

When a radio wave propagates in the environment, it is affected by the obstacles that it encounters. The most relevant propagation mechanisms that affect the propagation of a signal in a mobile communication system are reflection, scattering and diffraction [Rappaport, 2002, p.113].

- Reflection occurs when a wave impinges on a smooth surface whose dimensions are much larger than the signal wavelength λ. As a result of the reflection, the wave can lose part of its energy.
- Diffraction occurs when the path from the transmitter to the receiver is obstructed by an obstacle much larger than λ. According to the Hyugens' principle, every point at which the wave arrives becomes a source itself. Thus, even if the obstacle is in between the transmitter and the receiver, a secondary wave is formed that *bends* around the obstacle, being able to eventually reach the receiver.
- Scattering can occur when a wave impinges upon an obstacle of a size in the order of λ or smaller, resulting in its energy being spread in different directions. In practice, scattering can be produced by e.g. rough surfaces and small objects, such as street lights, signals and foliage.

The details of the effects of these phenomena can be calculated by solving the Maxwell equations with boundary conditions that express the electric properties of the obstacles. This calculation is, in general, difficult, and often it is not possible to know the electrical properties of the materials of the obstacles in detail. To overcome this difficulty, simplified theoretical models have been created that calculate approximations without needing to use the Maxwell equations directly. The most common models use ray-tracing techniques, in which the radio signals are approximated as planar waves represented by rays travelling in the direction of propagation of the wave. A clear LoS condition can then be modelled using one ray from the transmitter to the receiver. Diffraction around a corner can also be thought of as bending a ray. If we, for example, consider a reflection from the ground or a wall, we can add a second ray that bounces off the reflecting surface.

One example is the popular *two-ray model*, which considers that the most important components that take part in the communication process are the direct LoS and a single reflection from the ground [Goldsmith, 2005, p. 30]. Figure 2.1 shows the simulated RSS based on the two-ray model with a carrier signal of 2.4GHz, transmitter and receiver heights of 4 and 1 m respectively and perfect reflection. The resulting RSS predictions are oscillating, with an average decay rate similar to that of free space until a point called break point, which is calculated as the distance at which the last maximum occurs (d_c in the figure). Beyond that point, the oscillations stop and the power decays proportionally to d^{-4} [Rappaport, 2002, p. 124]. As we will see, this model has been used to predict the signal level in diverse scenarios, such as in rural and urban areas.

Scenarios in which a few rays suffice to approximate the effects of the radio propagation environment can be modelled using analytical deterministic formulas derived from theoretical considerations. Such is the case for communication systems with high antennas and a rather clear LoS, like long terrestrial links and

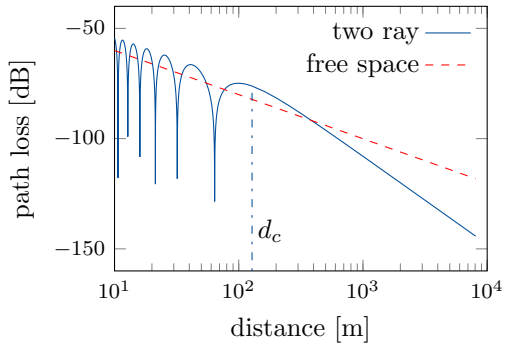


Figure 2.1: Two-ray model ($f = 2.4$GHz, $h_t = 4$m, $h_r = 1$m)

broadcasting systems. To aid in the development of these systems, propagation models were created to account for the curvature of the earth, the effects of the atmosphere and the large-scale features of the terrain, such as vegetation, the irregularity of terrain or the presence of large dominant obstacles (e.g. hills and other type of elevations) (see [Parsons, 2000, chapters 2 and 3]).

As the number of obstacles increases, the propagation environment becomes more and more complex due to the larger number of reflections, diffractions and scatterers. This constitutes what is known as a *multipath propagation environment*. In such scenarios, several copies of the original signal can arrive to the receiver with different amplitudes and phases, and even with shifted frequencies due to the dynamic nature of the environment. All of these components, called *multipath components*, are combined in the receiver and can distort the original signal. Multipath components arriving in phase with the original signal will increase the signal strength, whereas those in anti-phase will attenuate it.

Additionally, real environments are susceptible to random noise, interference and other hurdles, including mobile objects that change the characteristics of the channel over time. The result is a very complex propagation environment in which the particular environment being considered strongly influences the communication channel in a unique manner. Such a phenomenon cannot be easily modelled deterministically even with ray tracing techniques. If more realistic path loss predictions are required, then with the help of computers it would in principle be possible to use as many rays as desired. But the predictions created are then valid only for the particular site being considered, and do not give generic analytical models.

Instead of relying on theoretical considerations, sometimes it is easier and more practical to create empirical models. This type of models are constructed by fitting curves to field measurements. The validity of these models is then

limited to the environments and conditions in which the experiments took place. Their usage in other frequencies and/or environments is questionable and require further measurements for validation. It is also possible to find models that arise originally from theoretical considerations, and to which correction factors are added based on empirical observations. Nevertheless, the validity of these models is also restricted to the experimental conditions in which they have been tested.

2.2 Large-Scale Path Loss, Shadowing and Small-Scale Fading

The multipath propagation scenario has been studied thoroughly, as is the most natural type of scenario for built-up areas. It has been observed that the behaviour of the RSS as a function of distance in a multipath environment depends on the scale of the distance variation being considered. Three scales are considered: large, medium and small.

Traditionally, in the cellular network literature the concept of large-scale variations refers to variations of the distance in the order of hundreds of metres or even more. Changes of distance in that scale result in a global decrease of the RSS as a consequence of the spread of the energy. Models aiming to predict RSS changes at this scale, or in general the average change in the RSS with distance at a macroscopic level, are sometimes called *large-scale models* [Rappaport, 2002, p. 105].

When the distance variations being considered are in the same order of magnitude as the largest obstacles, the RSS can vary significantly around the mean value due to the gross effects of the obstacles. The fluctuation in RSS due to the overall effect of buildings and in general significant obstacles is called in the literature *large-scale fading*, *slow fading*, *shadow fading*, or simply *shadowing*.

The smallest scale of variation in distance is in the order of fractions of a wavelength of the carrier. At this scale, small changes in the T-R distance can result in relatively large variations in the RSS. These variations are a direct consequence of the alteration of phases and amplitudes of the multipath components that occur when the receiver moves small distances. This variation is usually called *small-scale fading*, or sometimes *fast fading*.

The existing literature often provides models for large-scale path loss, shadowing and fast fading independently, although some models consider several of them at the same time. This depends on the target scenario that the model has to represent. If, for example, we design a fixed long link between two distant regions using high antennas, most likely we can neglect multipath components other than those coming from clear reflections off large surfaces, such as the ground or other relevant surfaces. The main point of interest will then be the

average RSS, whereas the shadowing will not be of major significance. If, on the other hand, we design the base station of a cell in an urban environment with mobile terminals, then the fading effects will become very relevant. It might then be necessary to have a model of the shadowing, for example to calculate the percentage of coverage area of the cell and the outage probability ([Rappaport, 2002, ch. 4.9.3], [Goldsmith, 2005, chs. 2.9 and 2.10]).

2.3 Models for Large-Scale Path Loss

2.3.1 Large Unobstructed Links

Let us start by considering the propagation of waves in free and isotropic space. In such ideal conditions the signal propagates along a straight line, and can be modelled with a single ray. The relation between the transmitted and received power for a carrier wavelength λ and T-R distance d is given by the Friis formula [Friis, 1946]

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2, \tag{2.3}$$

where P_r is the available power at the receiver antenna terminals, P_t is the power delivered to the transmitter antenna terminals and G_t and G_r are the transmitter and receiver antenna gains, respectively. The received power decreases together with the square of the distance in accordance with the principle of energy conservation. This formula is valid for the far field of the transmitting antenna and for distances so that $d \gg \lambda$. The path loss in free space is then

$$PL[dB] = 10 \log \frac{P_t}{P_r} = -10 \log \left[\frac{G_t G_r \lambda^2}{(4\pi)^2 d^2}\right]. \tag{2.4}$$

The free space is an idealized environment that does not exist in reality. In practice, the path loss predicted by the Friis equation is a valid approximation for unobstructed environments with clear LoS conditions, such as in satellite communications and microwave LoS radio links [Rappaport, 2002, p.107].

Other typical applications requiring large-scale path loss models is long terrestrial links and broadcasting. Broadcasting systems aim at covering as large areas as possible, sometimes covering distances of 100 km or more. Hence, the transmitting antennas are located in high buildings transmitting the maximum allowed power. The receiving antennas are also placed on the roofs of the buildings, aiming at having as clear LoS as possible with the transmitting antenna. Similarly, in long terrestrial links the antennas are also placed in high positions, with a rather clear LoS. To aid in the development of these systems, propagation

models were created to account for the curvature of the earth, the effects of the atmosphere and the large-scale features of the terrain, such as vegetation, irregularities in the terrain or the presence of large dominant obstacles (e.g. hills and other types of elevation) (see [Parsons, 2000, chapters 2 and 3]).

2.3.2 Macrocells

The introduction of cellular mobile radios accelerated the appearance of methods to calculate the RSS in different types of built-up scenarios. In the initial phase the number of subscribers was expected to be small, and so the first models were developed for cells with radii of about 1 to 20 km or more [Bertoni, 1999, p. 24]. These cells are referred to as macrocells, and their base-station antenna is usually mounted on the roof of buildings or on high masts. The area covered by macrocells usually contains many buildings and small areas used for different purposes, all with their own electrical characteristics. The variability among them will in general be quite large, and the influence of particular buildings or regular areas have relatively little influence in the model. The major factors that affect the model are the global characteristics of the terrain.

The type of terrain covered by macrocells can vary significantly, each type having its own propagation particularities. Attending to the degree of construction in a particular area, we can find models for rural, suburban, urban and dense urban environments, each one with its own sub-classification [Parsons, 2000, ch. 4]. Usually there is no clear LoS between the base station antenna and the mobile unit, and the communication is achieved mostly through multiple diffractions around corners and rooftops and through buildings.

Many researchers around the world have measured the sector-average or median received power versus distance in different cities, proposing empirical propagation models for macrocells based on such measurements. One of the most influential studies is the work by Okumura *et al.* [T. Okumura and Fukuda, 1968]. Okumura and Fukuda conducted an extensive measurement campaign in which they measured the attenuation of a signal between a base station and a mobile receiver in Tokio for frequencies between 150 MHz and 1920 MHz and for distances from 1 to 100 km. The results were plotted in curves giving the median of the attenuation relative to propagation in free space.

A decade later, Hata used the measurements obtained by Okumura and Fukuda to propose formulas for the path loss measured in dB between isotropic antennas. These formulas are simple straight lines of the type $PL = -10 \log A + 10\, n\, \log d$, where the parameters A and n depend on design factors such as the base station's and subscriber's antenna heights and the type of city (size and type of terrain).

The measurements by Okumura and Fukuda together with the formulas pro-

posed by Hata have become a reference tool for the design of cellular systems, and they are explained in detail in standard text books related to radio propagation [Bertoni, 1999, Parsons, 2000, Rappaport, 2002, Goldsmith, 2005]. In the literature, sometimes they are simply referred to as the Okumura/Hata model. Later, the European Cooperative for Scientific and Technical Research (EURO-COST) formed the COST-231 committee to extend Hata's model up to 2GHz [COST 231, 1991].

The free space path loss, the Okumura/Hata and the COST-231 models can all be expressed in the following form

$$\bar{P}_r \propto P_t \left(\frac{d_0}{d} \right)^n, \tag{2.5}$$

where $\bar{P}_r$ is the average received power, n is a constant parameter fundamentally characteristic of the environment called path loss exponent, d is the T-R distance and d_0 is a reference distance, usually representing the antenna's far field above which the model is valid.

Equation (2.5) is sometimes referred to in the existing literature as the *simplified path loss model* or the *power law*. This relation is sometimes expressed in terms of the path loss instead of the transmitted and received powers, leading to the following expression [Rappaport, 2002, p. 138]:

$$\overline{PL}(d) \propto \left(\frac{d}{d_0} \right)^n,$$

where $\overline{PL}$ is the average path loss. From (2.5), adding explicitly the proportionality constant we can write

$$\bar{P}_r = K \; P_t \left(\frac{d_0}{d} \right)^n, \tag{2.6}$$

where K is unit-less and depends on the characteristics of the antenna and the average channel attenuation [Goldsmith, 2005, p. 41]. When measuring the power in dBm, the previous relation becomes

$$\bar{P}_r(d)[dBm] = P_t[dBm] + K[dB] - 10\, n \, \log\left(d/d_0\right), \tag{2.7}$$

which is the equation for a straight line. Given that $d = d_0$ then $P_r(d_0)[dBm] = P_t[dBm] + K[dB]$. This means that the term $P_t[dBm] + K[dB]$ can be interpreted as the received power at a distance d_0. At this point, we can rewrite (2.7) as

$$\bar{P}_r(d)[dBm] = P_{d_0}[dBm] - 10\, n \, \log\left(d/d_0\right), \tag{2.8}$$

where P_{d_0} is the intercept of the line, which itself is the received power at distance d_0. For brevity's sake, we will refer to it as the reference power through the rest of this thesis.

Equation (2.8) is also commonly known as the log-distance model, a name that makes explicit reference to the dependency of the received power on the logarithm (in base 10) of the distance. An equivalent expression for calculating the average path loss is [Rappaport, 2002, p. 138]

$$\overline{PL}(d)[dB] = \overline{PL}(d_0)[dB] + 10\, n\, \log(d/d_0). \tag{2.9}$$

The values of n, $\overline{PL}(d_0)$ (or equivalently P_{d_0}) and d_0 are usually determined either by using linear regression from experimental data or by some analytical formula derived from theoretical considerations.

2.3.3 Microcells

With a gradual increment of the number of subscribers the cell size had to be reduced to increase the system capacity. A new type of cell was introduced with a radii in the order of approximately 200 m to 2 km. This cell type is commonly called a microcell. To decrease the cell size, the base station antennas are typically mounted below the average rooftop and the transmitted power is decreased. Due to the difference in the antenna heights and cell sizes, the propagation conditions with respect to macrocells are substantially different. LoS propagation conditions are more likely since the base station antennas are closer to the street level. Additionally, the area covered by microcells contains much fewer buildings, and their particularities and geographical disposition can now play a much more important role [Whitteker, 1988]. As a consequence, it is more likely that the shape of the cell is not circular, and the applicability of radial models can sometimes be questioned.

A new wave of measurement campaigns was carried out during the 1980s and early 1990s to study the propagation in microcells in different types of terrains and under different LoS conditions. In [Xia et al., 1992], for example, measurements were made in both rural and urban environments. According to the observations, with a clear LoS the received signal behaved differently in two distinct near and far regions, [1] independently of the type of terrain. The RSS decays more rapidly in the far than in the near region, with decay rates of more and less than two respectively. The regions are separated by a breakpoint that can be calculated as the T-R distance at which the first Fresnel zone just touches the ground [Xia et al., 1992].[2] The two-ray model (see Figure 2.1) offered a good approximation of the observations. The difference between the rural and urban environments was

[1] Do not confuse this far region with the far field of a transmitting antenna.

[2] According to the diffraction theory, if the T-R distance is shorter than the break-point, it

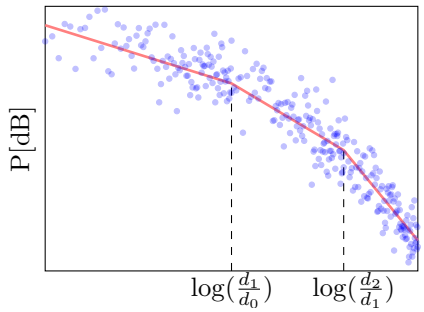


Figure 2.2: Multi-slope model with two breakpoints (three segments) and no discontinuities.

the presence of more fast fading in urban scenarios and the difference between the line slopes in the near and far regions. Interestingly, a slope of close to one was observed in the near region for rural environments, which is less than for free space. Using four rays instead of two resulted in better fast fading predictions in the suburban scenarios.

The pattern of the RSS behaving differently at different distance ranges has been observed in multiple scenarios, not only in microcells. In some cases more than two regions have been identified; therefore more than one break-point is necessary. Figure 2.2 represents one example of a model for three different regions. This type of model is usually called a *multi-slope* or *piecewise* model. The breakpoints and parameters of the lines can be adjusted using theoretical considerations or empirical data. For example, the two-ray model can be approximated via a piecewise model with two segments, the first of which has the slope of the free space model and the second one decays with the forth power of the distance [Rappaport, 2002, p. 124]. A complementary example can be found when considering hilly terrain, in which although the piecewise model can be used, the two-ray model does not satisfactorily explain the observations. In [Xia et al., 1992] the breakpoint was calculated as the distance to the top of the first hill, after which a dramatic increase in the slope was observed. Moreover, in some cases the fitted lines of the different regions do not intersect at the breakpoint distance, resulting in models with a small discontinuity. Once again, we see a fall-back to purely empirical approaches aiming at producing simple and usable models for complex environments.

can be considered that the ground does not obstruct significantly the signal, and the propagation can be assumed similar as in free space. With increasing T-R separations greater than the breakpoint, the first Fresnel zone gets blocked gradually by the ground, and therefore the decay rate of the signal will increase.

The effect of the geometry of the streets can also have a large impact on microcell models. For example, in areas with regular street grids, the propagation in directions similar to the T-R radial line is different than in circumferential directions. Har *et al.* studied the path loss for different types of receiver movements in a regular street grid: lateral, transversal, and staircase [Har et al., 1999]. These resulted in different path loss exponents and intercepts, all of which were dependent upon the height of the transmitting antenna with respect to the nearby buildings. The authors proposed independent formulas for each type of movement as well as a single formula with correction factors that can be used in low-rise environments with no LoS. The path loss exponent was roughly between two and seven. The cell shape was linear when the transmitter antenna was below the rooftop and in the middle of the blocks, whereas it was circular with extensions in the direction of LoS when the antenna was above rooftop.

The two-ray model has proven useful for explaining the observations in different types of multipath scenarios. In urban environments with very regular street grids, however, often there are other important multipath components that significantly affect the RSS, such as reflections from buildings. In such cases, increasing the number of rays used by the propagation model can help to better explain the observations. For example, the dielectric canyon model [Amitay, 1992] considers the propagation in rectilinear streets with tall buildings and/or row houses along both sides of the street and antennas close to the street level. The model uses ten rays taking into account single, double and up to triple reflections. Each reflection produces losses, and the components from paths with more than three reflections are ignored. This model explained well the rapid signal fluctuations observed when the receiver moves short distances.

2.3.4 Picocells

Continuing with the reduction of scale, picocells are usually designed to give coverage to buildings or floors within buildings. Indoor propagation has its own characteristics. The scale of the network is in the same order as the size of the building, and the construction details of the particular building being considered will deeply affect the propagation environment. Details such as the layout of the building and the construction materials become very relevant, and the variability of the environment related to the T-R distance is much greater than with macro and microcells. For example, the RSS can be strongly affected by factors such as doors being opened or closed or the position of the antennas within the buildings (e.g. at desk level vs. mounted on the ceiling) [Rappaport, 2002, ch. 4.11]. Propagation indoors can in fact be so different than outdoors that it actually serves as a classification criterion of propagation models.

Again, many measurement campaigns were carried out during the 1980s and

1990s to study the propagation channel indoors. In a classic survey, Hashemi distinguished between four different types of models [Hashemi, 1993]:

1. The log-distance model of equations (2.8) or (2.9) is widely used, mainly due to its simplicity and its successful application in other environments. In general, the path loss exponent is much larger than in outdoor environments. However, a very wide range of values have been reported, with values as low as 1.01 in corridors[3] [Rama Rao and Balachander, 2013].

2. Multi-slope models have also been used to describe the propagation indoors. For example, in [Akerberg, 1989] a model with four breakpoints is proposed based on measurements made in a multi-storey building.[4] Åkerberg suggests upper and lower limits for the path loss, with the variability being due to shadowing.

3. Attenuation factor models. The originality of these models lies in that attenuation factors corresponding to walls, floors and other large obstacles are added to the log-distance model, resulting in more accurate predictions of the path loss [Seidel and Rappaport, 1992, Skidmore et al., 1996, Durgin et al., 1998]. This requires having a floor-plan of the building and adds complexity to the model.

4. Free-space plus linear path loss model [Devasirvatham et al., 1990]. The motivation for this model is simple: the RSS decreases naturally with the distance just as in free space, and additionally it passes through many obstructions. The obstruction of each of the objects can be characterized with a fixed RSS decrease in dB. In the presence of many obstructions, the global effect can be modelled as a constant attenuation per metre of T-R separation.

2.4 Fading Models

The models presented in Section 2.3 predict the average RSS for a given T-R distance, but they do not give information about how the signal fluctuates around the mean in the area where it is averaged. As argued in Section 2.1, due to the complexity of propagation in multi-path environments many statistical models have been developed to explain the slow and fast fading.

Quite often, shadowing is modelled assuming that the ratio P_t/P_r is distributed according to a log-normal distribution. Because the logarithm of a

[3] Corridors can act as a waveguide, which explains the smaller decay rate than in free space.

[4] This model is sometimes referred to as the Ericsson multiple breakpoint model [Rappaport, 2002, p. 161]

log-normally distributed random variable is normally distributed, the path loss measured in dB is normally distributed. Consequently, the RSS is also normally distributed, as calculating the RSS from the path loss and the transmitted power only involves a subtraction (see (2.2)). Therefore, the PDF of the received power can be expressed as

$$p(P_r) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{1}{2}\frac{\left(P_r - \bar{P}_r(d)\right)^2}{\sigma^2}\right\}, \tag{2.10}$$

where $\bar{P}_r(d)$ is the average received power for a given T-R distance d, and for notation simplicity all of the powers are expressed in dBm. It is worth noting that this distribution is defined for P_r taking any real value, and therefore there can be cases in which $P_r > P_t$, which is not physically possible [Goldsmith, 2005, p. 43]. This is more likely to happen for values of $\bar{P}_r(d)$ close to P_t with relatively large values of σ. The previous model then better represents the real physical phenomena for $P_t \gg \bar{P}_r(d) + \sigma$.

The log-normal distribution explains well the effects of shadowing in many indoor and outdoor environments, such as residential areas, offices and supermarkets [Allsebrook and Parsons, 1977, Seidel and Rappaport, 1992, Erceg et al., 1999]. The traditional justification of shadowing log-normality found in textbooks is based on the hypothesis that the signal experiences numerous attenuations and distortions on its way to the receiver, which can be modelled as multiplicative factors. These can be considered random, as usually they affect differently and in an unstructured manner. When measuring the RSS in dB, the multiplication of factors becomes an addition and, by the Central Limit Theorem and for large enough number of factors, their sum tends to be normally distributed. However, this explanation has been questioned [Salo et al., 2005].

The envelope of a signal suffering fast fading is usually modelled using a Rician, Rayleigh or Nakagami distribution. These distributions are typically used to describe the observed temporal variability in the signal. In mobile communications, the mobility will produce temporal changes in the channel that cause signal variations in time. The Rayleigh distribution is used to describe the amplitude of a signal experiencing fast fading in the presence of multiple scatterers without a clear LoS component. The theoretical origin of the Rayleigh distribution resides in the assumption that the in-phase and quadrature components of the arriving signals are independently and normally distributed and that their phase is uniformly distributed. The power of the signal is proportional to the square of the amplitude, which is a summation of two squared Gaussians. Since the addition of k squared independently distributed Gaussians is a chi-squared distribution, the power is distributed as a chi-squared distribution with two degrees of freedom, which is an exponential distribution [Cavers, 2000].

If one of the components is clearly dominant, then the Rician distribution is

used instead to describe the amplitude of the received signal. As the dominant component fades away, the Rician degenerates into the Raileigh distribution.

2.5 The Log-Normal Model

Perhaps the most used RSS-distance model in scientific literature is a combination of the generalized path loss model and log-normal shadowing. This combination has become so popular that it is sometimes simply called the log-normal model. For notation simplicity and for the rest of the thesis, the power is assumed to be measured in dBm, even if not stated explicitly. Additionally, the dependence of the received power on the distance is omitted. Taking this into account, the log-normal model can be written as

$$P_r = P_{d_0} - 10\,n\,\log\left(d/d_0\right) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{2.11}$$

The resulting P_r is a normally distributed random variable whose density is given by (2.10) with $\bar{P}_r(d) = P_{d_0} - 10\,n\,\log\left(d/d_0\right)$.

The path loss exponent measures the rate of decay of the RSS with the log-distance. Generally, it is expected that n is greater in more cluttered environments, as the signal is interfered with by more obstacles and, on average, will arrive to the receiver with less power compared to less cluttered ones. The standard deviation, σ, is a measure of the uncertainty of the model for RSS predictions, and the larger it is the less information the model gives about the RSS. The main factors that contribute to increase σ indoors are shadowing and small-scale fading. These are very difficult to predict, as they depend heavily on the particular layout of the environment and the electrical properties of all the obstacles.

Seidel and Rappaport conducted an extensive measurement campaign in the early 1990s and provided deterministic values for n and σ for different types of buildings [Seidel and Rappaport, 1992]. For their measurements, they used the same transmitter and receiver pair of high quality radios equipped with omni-directional antennas. With respect to P_{d_0}, they used a reference distance d_0 of 1 m, for which the authors considered that the received power was affected only by the propagation in free space. Thus, they treated P_{d_0} as a constant that basically depends only on the equipment used for the experiments and not on the environment itself. This procedure of calculating P_{d_0} is usually done for systems with radios that cover large areas of a building.

2.6 On the Usability of the Log-Normal Model in WSNs

The log-normal model is a simple radial model inspired by the Friis formula whose parameters are chosen directly from experimental data. Its form is simple in that it hides all the complex details of the radio-propagation phenomena and predicts the behaviour of the signal strength using an elementary equation. The theory suggests the shape of its mean function, which is a straight line, and practical measurements together with mathematical convenience advocate for a normally distributed random variable that accounts for the shadow fading and other sources of uncertainty. Researchers then decide whether the model appropriately explains their observations or not, and if it does, they fit the line minimizing the errors with respect to the measurements. All in all, it is a mathematical empirical over-simplified and very convenient model, which might or might not work depending on the experimental conditions.

Anyhow, it certainly is a *de-facto* standard in scientific literature, and it has demonstrated to be useful in many studies. Therefore, its use is ultimately justified by its utility. Nevertheless, when using it one should be aware of its structural limitations, which can cause the model to produce systematically biased RSS predictions. This will certainly impact localization algorithms that rely on the model, which in turn can produce biased position estimates. In this section we provide some thoughts on two such limitations whose effects on the bias arise especially when the log-normal model is used in WSNs: radiality and linearity.

2.6.1 Radiality

The log-normal model is radial, meaning that its predictions depend only on the T-R distance. This constraint causes radial models work optimally in environments with no obstructions and when using omnidirectional antennas. In real WSN deployments, the presence of obstacles is almost unavoidable and the antennas used are far from ideal. These facts represent challenges for radial models. Their impact on the model's suitability can be explained in terms of the (relative) homogeneity of the environment and the omnidirectionality of the antennas.

Homogeneity of the Environment

Under the presence of obstacles, their geographical distribution, size and electrical characteristics impact the efficacy of the model at predicting the RSS. One main factor that makes radial models suitable for use is thus the homogeneity of the propagation environment. Homogeneous environments do not favour any special direction of propagation on a macroscopic level. As a result, the net effect of the

obstacles is a decay in the RSS as the T-R distance increases independently of the direction. On the other hand, inhomogeneous environments are more likely to produce favourable directions of propagation in the aggregate scale, making radial models less appropriate as this effect becomes more important.

The concept of homogeneity with respect to the propagation environment is relative to each scenario and is influenced by its scale. For example, we have argued that in macrocells the environment can be seen as randomly unstructured and therefore relatively homogeneous at the macroscopic level. In microcells, the relative importance of individual buildings within the area of interest (AoI) increases, and the propagation environment can usually be considered more structured than in macrocells. In picocells the scale of the network is even smaller, and often the obstacles are elements or parts of a building rather than entire buildings. Furthermore, the indoor environment can often be seen as structured due to the presence of infrastructure that facilitates propagation in certain directions (e.g. corridors). This increasing degree of structuredness implies that the relative homogeneity of the environment decreases. The usability of any radial model then becomes more and more questionable and dependent on the scale of the cell as well as on the particular distribution of the individual obstacles and their materials. Ultimately, whether or not a radial model is appropriate can become a decision that has to be be made on a *per-case* basis.

WSNs typically operate on even smaller scales. For example, the maximum T-R distance achieved with our nodes is in the order of 20 m indoors. In such scales, a single dominant obstacle (e.g. a metallic wall) can notably affect the propagation in the whole AoI. In general, it can be argued that the propagation conditions can be heavily affected by the specific location of the nodes within the environment, simply because of the different relative distribution of the obstacles. Therefore, in such a small-scale networks the local characteristics of the environment can have a profound impact on the propagation channel and on the validity of the model.

Whether a radial model will or will not be useful depends on how well it can predict the observations within the AoI, and ultimately on the requirements of the application in which will be used. The fact that an RSS model is radial does not necessarily imply that the AoI should be circular. Certainly routers can be installed on the inner side of the outer walls of a building or even in corners giving service to only certain sectors, and indeed we can still use radial models for those areas. The key idea is simply how well the model can predict the measurements in relation to the application's requirements.

In this thesis, the application is the localization of nodes from WSNs using the set of RSS measurements obtained from packets sent by a mobile robot that follows a random trajectory. The overall AoI is essentially the area covered by the robot, which is the area for which the model has to explain the observations.

The position of the nodes with respect to the AoI necessarily differs from node to node: in the best-case scenario[5] a node will be in a central place within the AoI and the robot will send beacons from positions all around it. In the worst case, a node will be completely outside the AoI. Still, the task of the model is to predict as well as possible the observations from all the nodes.

Because of the relative inhomogeneity of indoor environments given the scale of our network, the particular position of the nodes within the setup will condition the robot-node communication channel uniquely for each node. Due to these heterogeneities, a single log-normal model will have to make global compromises in an effort to explain the observations from all the nodes. Instead of using one global CM, in the following chapter we will explore the possibility of using one model for each node, a paradigm that we have denoted as using IMs. An IM needs to explain only the observations made by its associated node. This spatial specialization results in better overall RSS predictions for all nodes compared to using a single CM, which can result in better position estimates. It does not, however, completely solve the intrinsic misspecification problem of radial models when used in inhomogeneous environments.

Antenna's Omni-directionality

Because of its radiality, the log-normal model assumes that the antennas used are perfectly omnidirectional. WSNs, however, are typically formed by low-cost small nodes whose antennas are far from omnidirectional. The RSS then becomes a function of the relative T-R angle.

The inability of the log-normal model to explain this results once more in biased RSS predictions which are globally compensated for by the model. The net effect can be seen in the model parameters, which have to change to accommodate measurements for which the model is not explicitly prepared. Ultimately, this will contribute to the bias of position estimators relying on the model.

In the next chapter we will present a quantitative study of how the directionality of the antennas can affect the model parameters. The goal is to quantify this effect, which eventually will allow us to ponder its significance.

2.6.2 Linearity

Another structural limitation of the log-normal model is its linearity with respect to the log-distance. As we have seen in Section 2.2, in many cluttered environments the observations have been found to follow a pattern in which the decay rate of the RSS increases along with the distance.

[5] In the sense of geometric dilution of precision (GDoP).

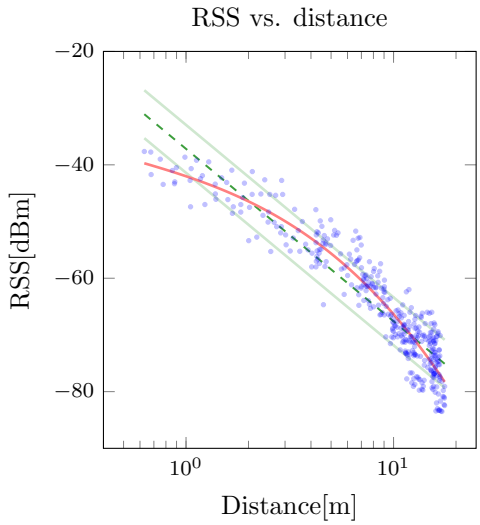


Figure 2.3: RSS-distance scatter plot for a set of measurements obtained in the office. The dashed green line corresponds to a log-normal model fitted to the data, and the solid red line to a third-order polynomial on the log-distance.

We have observed a similar pattern in our experimental scenarios, especially with measurements collected in the office. Figure 2.3 presents a scatter plot of RSS-distance measurement pairs associated with one of the nodes located in our office. We can observe that the trend of the cloud is slightly curved, and therefore a straight line is not the best function to be fitted to the data (dashed green line). For a comparison we also plot a fit of a third-order polynomial on the log-distance[6] (solid red line), which better explains the observed trend. The linear predictor has three differentiated areas in terms of its bias with respect to the third order polynomial: one with a positive bias for small distances, a second one with a negative bias and a third one with again a positive bias. The linear model compensates for this globally, creating a model that is systematically biased at all distances. The third-order polynomial also compensates globally for bias at different distances, but the overall average predictions are more accurate and precise.

In order to overcome this type of systematic bias, in later chapters we will also try to use other radial models more flexible than the log-normal. This effort can be seen as a further specialization of the model after switching to using IMs, with the aim of also producing better RSS estimates.

[6]The reason for choosing a third-order polynomial is explained in Section 5.6

Chapter 3

Variability of the Log-Normal Model Parameters

Contents

Selecting the appropriate model for the RSS is fundamental for localization algorithms, as their performance depends heavily on the accuracy of the model predictions. Once we have chosen to use the log-normal model, the value of its parameters must still be properly selected so that the model explains real observations as faithfully as possible.

The utility of the log-normal model has been corroborated by many studies in different types of environments. As we have seen in the previous chapter, especially in the late 1980s and early 1990s considerable efforts were made to characterize the indoor propagation channel, and many of those studies are now considered very influential classic references. We must not forget, however, that those campaigns aimed to evaluate the attenuation of the communication channel, and thus were conducted using professional equipment trying to minimize the effect of the HW used. Moreover, the measurements were typically obtained in entire buildings or floors, with maximum distances reaching as much as 100 m.

Real WSN deployments usually encounter experimental conditions with significant differences. First, the maximum transmitting range of the radios is substantially smaller, typically covering a few rooms. At this scale, the closer obstacles acquire more relative importance and will have a greater impact in the model. Thus, the local radio environment can vary more among the nodes, as it depends on the particular conditions of the environment close to the node in question. Second, instead of a pair of professional radios, WSNs are composed of many nodes equipped with low-cost transceivers, all slightly different, and non-omnidirectional antennas. As a consequence, the nodes will transmit and measure different power even in the same experimental conditions.

Empirical models identified using real RSS measurements will then be affected by the HW and environmental variability, and this effect will be seen in the model parameters. This raises some practical questions regarding the identification of empirical models. First, which nodes should be used to identify the model? And second, in what positions should we place the nodes when doing the measurements? The importance of these questions is ultimately determined by the application that the model is going to be used for, which in this thesis is localization. This raises additional questions. How does this variability affect the localization performance? Is this variability relevant, or is it something that we can ignore? And if we find out that it is relevant, what can we do about it?

This chapter is dedicated to studying the variability in the model parameters, leaving an assessment of how it impacts position estimation until Chapter 5. Section 3.1 will discuss how we identify the model. Then, Section 3.2 shows the variability in the model parameters depending on the nodes used to identify the models. In sections 3.3 and 3.4, we elaborate on the effects of the HW and the environment and how they influence this variability. Finally, Section 3.5 presents the correlation between the parameters, showing their interdependence.

3.1 Model Identification

From now onwards, and for the rest of this thesis, for notational simplicity we will use the letters z and z_0 to denote the RSS and and the reference power respectively. Using this, the log-normal model (2.11) becomes

$$z = \bar{z} + \epsilon \tag{3.1}$$

$$\bar{z} = z_0 - 10\, n \log\left(d/d_0\right) \tag{3.2}$$

$$\epsilon \sim N(0, \sigma^2).$$

The identification of this model involves making measurements of the RSS at different distances in the area of interest, and then choosing the appropriate values for the parameters so that the model explains the observations as best as possible. The optimality criteria is usually the LSE error between the predicted and the observed values. The log-normal model depends on four parameters: d_0, z_0, n and σ. In this section we will discuss how to choose these parameters.

As we have seen, the generalized log-distance model is valid only for distances within the far field of the transmitting antenna, and so it is the log-normal model. Usually d_0 is chosen as distance at which the far field region begins, and thus the model is said to be valid for $d > d_0$. Typical values at which the far field region is assumed to start are 1-10 m indoors and 10-100 m outdoors [Goldsmith, 2005, p. 41]. Sometimes d_0 is simply chosen within the far field. In general, the method used to calculate z_0 determines the selection of d_0.

In some of the classic studies involving the log-distance model, z_0 is equated to the received power at the reference distance considering ideal conditions of isotropic radiators and free-space propagation. This can be calculated using the Friis formula of (2.3), assuming that the antenna gains, transmitted power and losses in the cables and connectors are well known. The Friis equation is itself valid only for T-R distances within the far field of the transmitting antenna, and thus it is natural to choose d_0 within that range. Once z_0 is known, n can be estimated using least squares [Howard and Pahlavan, 1990, Seidel and Rappaport, 1991, Seidel and Rappaport, 1992, Durgin et al., 1998, Erceg et al., 1999].

A more practical method is to calculate z_0 as the average of the received power measured at different points at the reference distance. This does not require having detailed information about the equipment being used. In this case, it is especially important to choose d_0 within the far field where the model is valid. Again, n can then be estimated using least squares.

A third alternative is to use linear regression to estimate z_0 together with n [Ott and Plitkins, 1978, de Toledo and Turkmani, 1992, Erceg et al., 1999, Algans et al., 2002, Dabin et al., 2003]. We refer to this method as the full linear regression. In this case, the selection of d_0 will affect the value of z_0, but not the predicted values of the RSS. From a purely geometrical perspective, the pair

(d_0, z_0) corresponds simply to a point through which a straight line passes, and this line is fit to empirical data using linear regression.

Having three possible methods to calculate z_0 it is worthwhile to consider the differences and the implications of choosing one or another. The aim in any case is to fit a straight line to data gathered in experiments, and this implies calculating its slope and one point through which it passes. For the sake of brevity, we call this point the reference point.

The methods that use the Friis formula and that measure the average power at the reference distance both fix the reference point and then calculate n using least squares with only one degree of freedom. Thus, only the slope can be changed to fit the line to the data. Alternatively, the full linear regression method calculates both the reference point and n using LS. If the reference point calculated using either of the first two methods coincides with the reference point calculated using the full linear regression, then the LSE will be equal in all cases. However, if for any reason the reference point is different, then the LSE will be smaller when using the full linear regression. Additionally, the resulting standard deviation will be larger, as it has to be inflated to accommodate the extra errors due to reducing the degrees of freedom of the linear regression.

In macro, micro and picocells where the generalized path loss model has been adopted, typically the first two methods have been used. The justification for using the first one is that the environment does not contain obstacles at distances $d < d_0$, and therefore the path loss can be attributed to free space propagation only [Seidel and Rappaport, 1992]. The justification for using the second method is simply that the model is a straight line and that we need only to specify one of its points. The ultimate justification of either is simply that *it works*, in the sense that the model properly represents the observations [Erceg et al., 1999]. In any case, we must not forget that these classical studies aim at creating channel models, and therefore the effect of the HW is minimized during the experiments. Usually the same T-R pair of professional radios is used during the experiment, and their characteristics are well known.

In WSN the experimental conditions can be quite different than those encountered in these classical studies, and these conditions determine that the method typically used is the full linear regression [Savvides et al., 2001, Hightower et al., 2001]. First, communication occurs among many non-professional, low-cost nodes whose radio characteristics are not well known and may even vary from node to node[Hightower et al., 2001, Lymberopoulos et al., 2006, Nguyen et al., 2011]. Second, as discussed in the previous chapter, the RSS-distance scatter plot can often be non-linear (see Figure 2.3). In these cases, calculating z_0 using the Friis formula and subsequently n using LS will result in a worse model fit (larger LSE) than a full linear regression. For all of these reasons, in this thesis we will use a full linear regression to estimate the model parameters.

Let us denote $\mathbf{z} = (z_1, \ldots, z_K)^t$ the vector of K random RSS observations made at T-R distances d_k, where $k = \{1, \ldots, K\}$ and the superscript t denote the transpose. Under the model (3.1) we can write

$$\underbrace{\begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_K \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} -10\log(d_1/d_0) & 1 \\ -10\log(d_2/d_0) & 1 \\ \vdots & \vdots \\ -10\log(d_K/d_0) & 1 \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} n \\ z_0 \end{pmatrix}}_{\boldsymbol{\theta}} + \underbrace{\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_K \end{pmatrix}}_{\boldsymbol{\epsilon}} = \mathbf{A}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \tag{3.3}$$

where in this chapter we chose $d_0 = 5$ m. A full linear regression involves minimizing the sum of squared errors

$$J(\boldsymbol{\theta}) = (\mathbf{z} - \mathbf{A}\boldsymbol{\theta})^t(\mathbf{z} - \mathbf{A}\boldsymbol{\theta}). \tag{3.4}$$

The solution that minimizes (3.4) is the OLS estimator

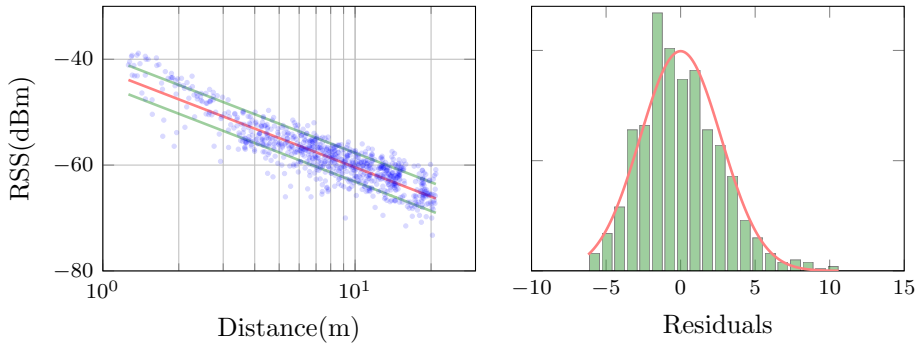
$$\hat{\boldsymbol{\theta}} = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\,\mathbf{z}. \tag{3.5}$$

We still need to calculate the parameter σ in order to completely define the model. This can be estimated using the sample standard deviation of the residuals

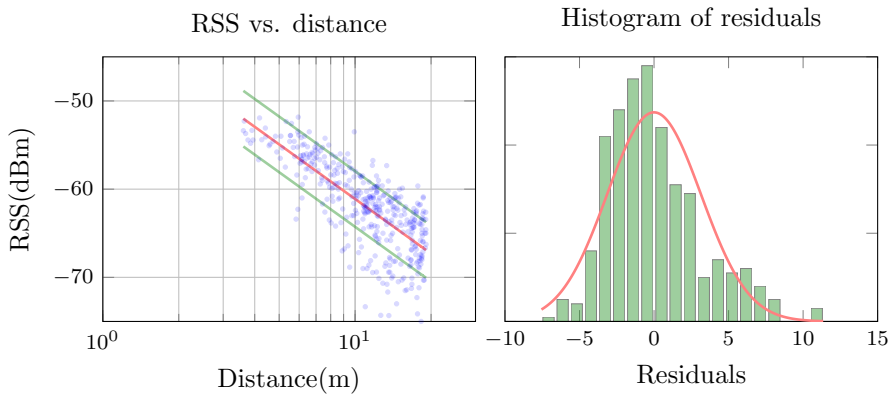
$$\hat{\sigma} = \sqrt{\frac{1}{N-2}J(\hat{\boldsymbol{\theta}})}. \tag{3.6}$$

The random variable ϵ has been used traditionally to model the fading. However, if the true RSS mean value is not linear in the log-distance, (3.6) makes it clear that $\hat{\sigma}$ will automatically increase as a consequence of the increased differences between the predicted and measured RSS. Therefore, ϵ accounts naturally for the shadowing and, more generally, for any other source of model uncertainty.

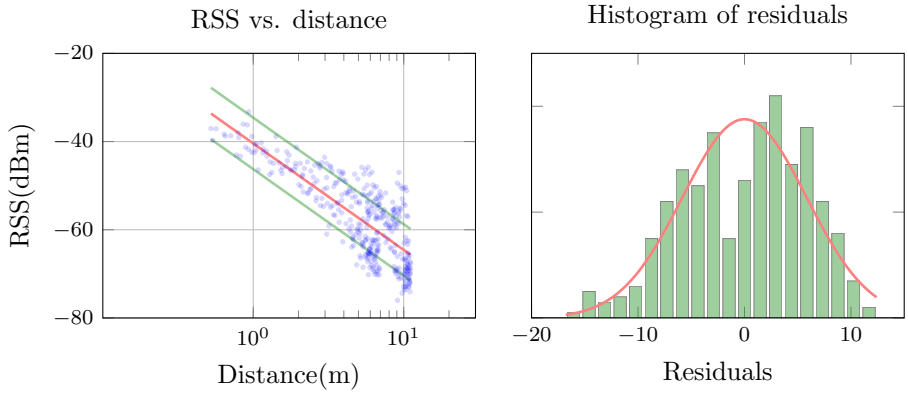
Figure 3.1 shows examples of typical RSS-distance scatter plots and residual histograms together with their associated model for measurements collected by one node in each of our three environments. Panel 3.1a shows a typical example for nodes in the basketball field. As we can see, the log-distance model is fit reasonably well to the trend of the RSS. The residuals are also close to being normally distributed, although in general they showed a tendency to be slightly skewed towards the negative RSS values. In the lobby the behaviour was more heterogeneous. In general the log-distance model was still a good predictor of the radial average, but the residuals were often clearly not normally distributed. We observed the same type of skewness as in the basketball field, but often more accentuated. Panel 3.1b shows a representative example. Panel 3.1c shows an example of a model from the office. We can observe a tendency of the slope of the average and the standard deviation to increase together with the distance. The residuals were rarely normally distributed, and often also multi-modal.



(a) Basketball field.



(b) Lobby.



(c) Office.

Figure 3.1: Examples of models identified in the different environments.

3.2 Variability of the Model Parameters

As argued in Section 2.6, in real WSN deployments the relative inhomogeneity of the environment due to the presence of obstacles can create unique propagation characteristics for each pair of communicating nodes depending on their specific location. Additionally, these networks are typically formed by many low-cost nodes, which, due to limitations in the manufacturing process, have slightly different radios even if they are the same model and from the same manufacturer. Moreover, the orientation of the nodes also affects the RSS, even when so-called omnidirectional antennas are used, as pure omnidirectional antennas do not exist. All these factors affect the model parameters.

The problem of dealing with differences in HW is sometimes referred to as the hardware variance problem in the existing literature. When the nodes are clearly different (different model and/or manufacturer), it is intuitive and clear that the differences will affect the RSS. How to deal with this problem is under active research in the area of fingerprinting localization ([Kjærgaard and Munk, 2008, Yang and Chen, 2009, Dong et al., 2009, Nguyen et al., 2011]).

In the case of WSNs, being the nodes from the same manufacturer and commercial model, these small differences are usually ignored. Typically, the same RSS model is used for all the nodes, and no special measures are adopted to tackle the variability. Moreover, the models are identified using data collected with the same pair of nodes. The aim of this section is to provide an example of the variability of the model parameters that can be observed in real deployments depending on the particular pair of nodes used to identify the model and their position. This variability is intrinsically related to the HW variance problem in WSNs and to the relative inhomogeneity of the environment, relations that will be discussed in later sections. Our study will not, however, include the effects of such factors as temperature and humidity. From now on, any reference to environmental factors refers to the spatial arrangement of the obstacles and their electrical properties.

Typically, one RSS-distance model is used by all the nodes in a network to predict RSS measurements. We denoted this type of model as a **common model** (CM). A site-specific CM is then a model whose aim is to predict the RSS optimally in a target environment or AoI for all the nodes involved.

Site-specific CMs are usually identified using RSS measurements obtained using a pair of nodes in the AoI, keeping one in a fixed position and placing the other at different locations. This is the typical procedure followed for example in cellular networks, in which the fixed node is the base station and the other one is the mobile terminal. In WSN, having many nodes to choose from and knowing that they are all slightly different, we can ask ourselves which pair should be used for the training data collection. Because the model is to explain observations from

all the nodes, ultimately it would be desirable to estimate the model parameters using data from all the nodes.

In our experiments we have N nodes in three different environments collecting RSS measurements in fixed positions of packets sent by a mobile robot. In each environment we have data collected in three tests in which the robot follows different trajectories (see Section 1.4). The best site-specific CM that optimally explains[1] all the RSS measurements collected by all N nodes during a single robot trajectory is the one whose parameters are estimated using all the N datasets (one per node) associated with that test. In the following, we will refer to this model as the **reference model** for the given robot trajectory, and we will use it as a reference to compare other models.

An alternative to using a CM to explain the measurements from all nodes is to dedicate one model to predicting the observations from each of the nodes individually. These models are be identified using only the measurements collected by its corresponding node. We denoted this type of model as an **individual model** (IM). Because the nodes are different and are placed in different positions, studying the parameter variability among IMs will give us an idea of the aggregate impact of the HW's variability and environmental inhomogeneity on the model parameters.

Figure 3.2 presents the parameter values of the IMs and reference models identified using data collected in one of the robot trajectories for each environment. The small dots represent the parameter values of the N IMs, and the curves their empirical distributions. These were calculated using kernel density estimation as described in [Botev et al., 2010].[2] The thick dots represent the parameter values of the reference models. As shown, the parameter values differ considerably among the three environments. In addition, for IMs they can be remarkably different even within the same environment, as indicated by their dispersion.

Figure 3.3 shows the same results as Figure 3.2, but for the data collected in the three robot trajectories for each environment. IM parameter values associated with different robot trajectories are presented as dots in different horizontal lines. The value of the reference distance was chosen as $d_0 = 5$ in all cases. Table 3.1 presents the main statistics of the distributions and the parameter values of the reference models. As stated earlier, these results show the aggregate impact of the HW and environmental inhomogeneity on the model parameters. Due to the complex interaction among the causes, accurately quantifying their individual contributions is quite complicated. We proceed now to argue qualitatively about the effects of both.

[1] In the LSE sense.

[2] Code available at `http://www.mathworks.se/matlabcentral/fileexchange/17204`, retrieved in May 2012.

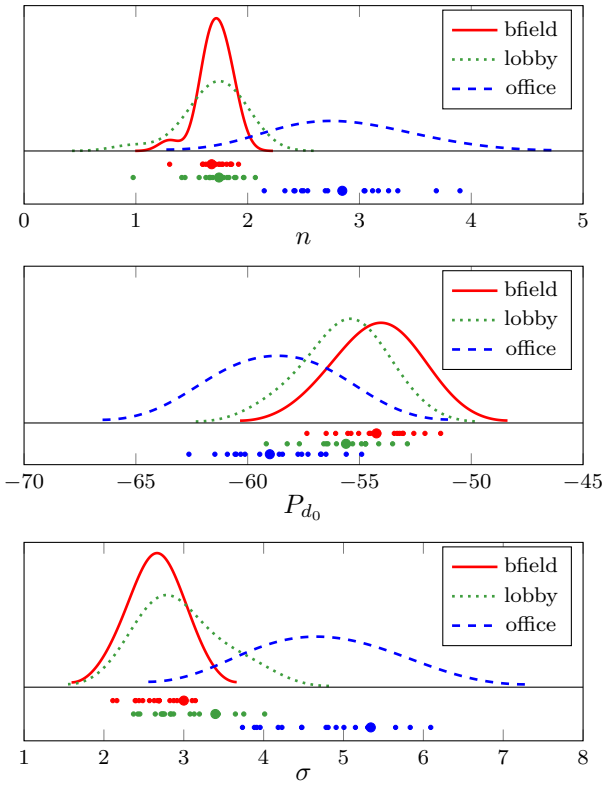


Figure 3.2: Reference (thick dots) and individual (thin dots) model parameters in the three environments for one robot trajectory ($d_0 = 5$ m).

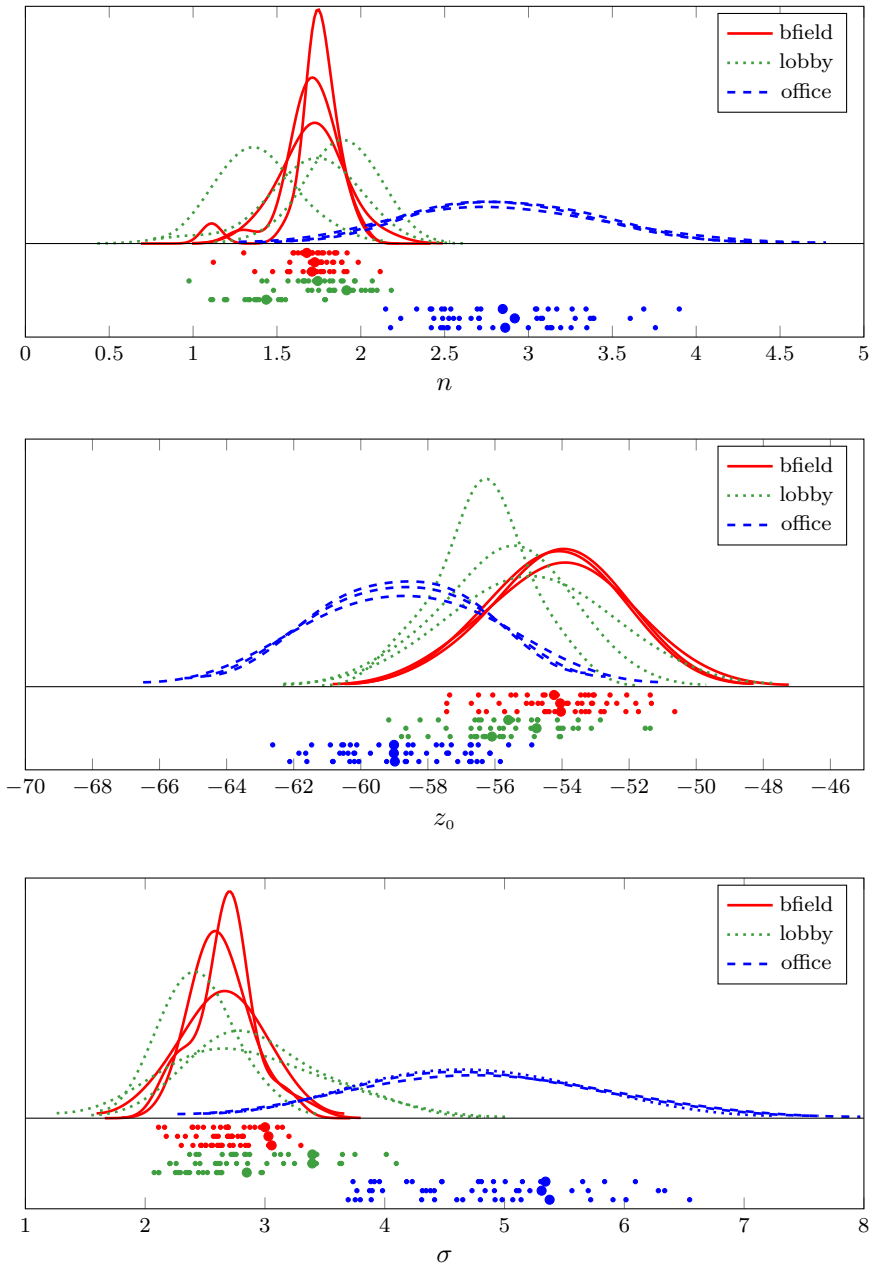


Figure 3.3: Reference (thick dots) and individual (thin dots) model parameters in the three environments for the three robot trajectories ($d_0 = 5$ m).

Table 3.1: Mean and standard deviation of the distributions from Fig. 3.3, and parameters of the reference models (rm)

(a) Trajectory 1

		Bfield	**Lobby**	**Office**
n	mean	1.71	1.70	2.87
	std	0.14	0.25	0.48
	rm	1.68	1.74	2.85
z_0	mean	-54.19	-55.67	-58.66
	std	1.54	1.60	2.18
	rm	-54.24	-55.60	-59.00
σ	mean	2.65	2.98	4.76
	std	0.28	0.47	0.71
	rm	3.00	3.40	5.34

(b) Trajectory 2

		Bfield	**Lobby**	**Office**
n	mean	1.74	1.89	2.89
	std	0.17	0.17	0.40
	rm	1.73	1.91	2.92
z_0	mean	-54.07	-54.86	-58.80
	std	1.58	1.92	1.87
	rm	-54.05	-54.76	-59.01
σ	mean	2.67	2.88	4.82
	std	0.25	0.55	0.79
	rm	3.03	3.39	5.31

(c) Trajectory 3

		Bfield	**Lobby**	**Office**
n	mean	1.72	1.39	2.84
	std	0.17	0.20	0.41
	rm	1.71	1.43	2.86
z_0	mean	-54.02	-56.32	-58.76
	std	1.72	1.28	1.97
	rm	-54.03	-56.09	-58.98
σ	mean	2.66	2.46	4.86
	std	0.23	0.26	0.83
	rm	3.05	2.85	5.38

3.3 Effect of the HW

Consider a perfectly static, homogeneous and isotropic environment, like free space, and a group of perfectly equal nodes with perfect isotropic antennas. In such an ideal scenario, the values of all the model parameters would be equal for all the nodes and their densities would be Dirac delta functions. The path-loss exponent would depend solely on the environment, and in particular, in free space it would be exactly equal to two. The reference power z_0 would depend only on HW-related factors: the transmitted power, the gains of the transmission lines and antennas, etc. Finally, σ would be small, and mainly due to thermal noise.

Now, consider using instead real nodes with small HW differences in the circuits, cables and connectors, and in general any internal circuitry that processes the RSS, but still with perfect omnidirectional antennas and in the ideal space. The net effect of these sources of variability will be a different transmitted and received power among the nodes, which will be seen in every RSS packet independently of the T-R distance.[3] Because of this independence, the path-loss exponent is not affected, and only z_0 will be influenced. As a result, the dispersion of the PDF of z_0 will increase. Following this line of reasoning, in practice the HW factors are considered to affect only z_0 and not the path-loss exponent. On the other hand, the path-loss exponent is usually assumed to be dependent on environmental factors only.

With respect to Figure 3.3 and z_0, we observe the following. First, the average depends on the environment. This is clearly related to the fact that we use $d_0 = 5$ m as the reference distance, distance at which the RSS is different in each environment due to the different values of n. Second, from a visual standpoint the dispersion of the empirical distributions of z_0 change relatively less among environments than for n and σ. This is confirmed by the statistics presented in Table 3.1, where we see that the relative variation of the standard deviations when changing environments is larger for n and σ compared to z_0. This suggests that z_0 is less affected directly by the environment than the other model parameters, which is in agreement with general practice.

Still, there is some variability that cannot be explained on the basis of HW effects only. First, although not being the main affecting factor, the change of environment does change the dispersion of z_0 with a clear pattern: it increases as the environment becomes more cluttered (see std. values in Table 3.1). Second, values associated with the same IM are different even among robot trajectories within the same environment. To this effect, compare in Figure 3.3 the dots presented along different lines belonging to the same environment. These re-

[3] Studies in which the effects of transmitter and receiver variability on the signal strength are specifically measured are for example [Hightower et al., 2001], [Lymberopoulos et al., 2006] and [Fang et al., 2010].

latively minor variations can be partially explained by the non-ideality of the antennas, the environmental inhomogeneity and the linear structural limitation of the log-normal model, issues that are discussed in the following sections.

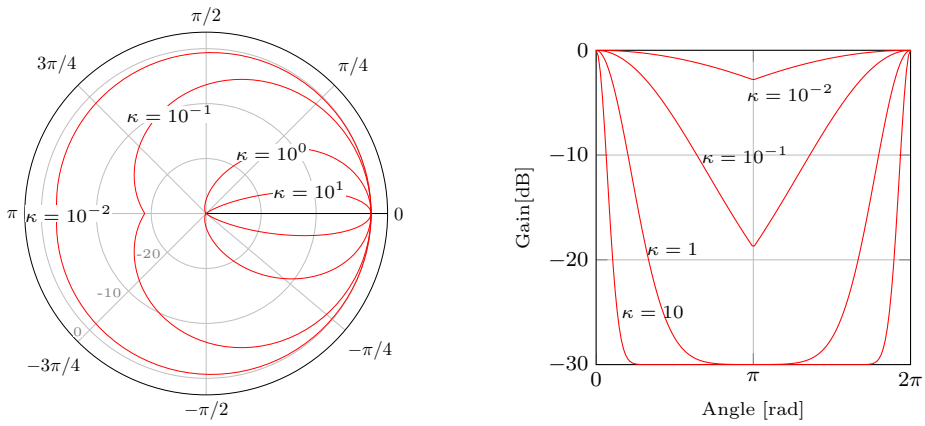
Antenna Directivity

Let us return for a moment to our semi-ideal setup with different nodes and a perfectly homogeneous and isotropic environment, and let us add non-perfectly omnidirectional antennas, making it HW-wise more realistic. The received power will now depend not only on the relative T-R distance, but also on the orientation of both, the transmitter and receiver. Then, a mobile receiver will in general not measure the same RSS in points located at the same distances. Therefore, in principle any model parameter could experience a change, although obviously we expect some parameters to be more affected than others. One immediate consequence is that σ has to be increased to fit the observed data. This increase is not necessarily the same in all of the IMs, and therefore the dispersion of the empirical distribution of σ will also increase. The effect of the antenna directivity on z_0 and n are not so straightforward to evaluate in a general scenario; however, following the previous line of reasoning, we expect that their distribution will also be more disperse. This could in part explain the variability of z_0 which could not be attributed exclusively to the other HW-related factors.

The previous paragraph implies that, although not to the same extent, the directivity of the antennas can influence all of the model parameters. We will now seek to validate this statement using our experimental data. Quantifying this effect is very complicated in practice, as it would require being able to repeat the same experiments in the same locations with different directional antennas with known radiation pattern. Instead, we will incorporate offline the effect of different simulated antennas as if they were mounted on the robot during our experiments, and then we will compare the results.

The procedure is as follows. Imagine that we have an antenna with known pattern whose effect we want to evaluate. We know the position and orientation of the robot at all times together with the position of all the nodes, and thus we can calculate the relative node-robot angle for the RSS from all the nodes. Using this information, we can calculate the antenna gain that would correspond to each measurement had the antenna been present during the experiment. We then add this gain to the respective observations. We do this for all the different antennas that we want to evaluate.

The previous procedure is not strictly equivalent to using a real antenna in practice. The main reason is that it implicitly assumes that the signals arrive to the receiver following only one path, the straight line connecting transmitter and receiver, while it ignores the different antenna gains for the different multi-path components. Our aim, however, is simply to obtain a first impression with



(a) Polar plot.

(b) Gain vs. angle plot.

Figure 3.4: Antenna pattern for different values of κ.

respect to how much the antenna directivity can affect the model parameters, and not to accurately quantify this effect. We then consider that the proposed method is suitable for our purposes.

Instead of considering completely different antennas, we create one virtual antenna with a directivity that can be parametrically controlled. Our virtual antenna has the following radiation pattern:

$$g = g_{\mathrm{m}} + (g_{\mathrm{M}} - g_{\mathrm{m}})e^{-\kappa\phi^2}, \ \phi \in \{-\pi, \pi\}, \tag{3.7}$$

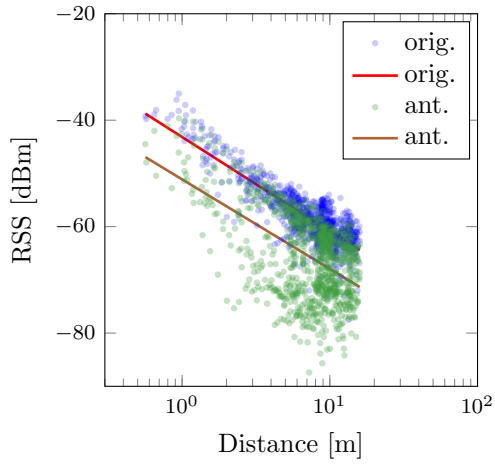
where g is the gain in dB, g_{M} and g_{m} are the absolute maximum and minimum gains of the antenna, ϕ is the relative angle and κ is the parameter that controls the directivity. Figure 3.4 show the radiation patterns for fixed values of $g_{\mathrm{M}} = 0$ dB and $g_{\mathrm{m}} = -30$ dB, and κ varying from 10^{-2} to 10^{1}. As we can see, the pattern becomes more homogeneous as κ decreases. In the limit when $\kappa = 0$, the antenna is perfectly omnidirectional with a gain of 0 dB, and thus it will have no effect whatsoever. As the value of κ increases, the antenna focuses more in directions close to zero degrees. Note that, because the gain will always remain between g_{M} and g_{m}, the radiation pattern does not have any null, and therefore our antenna never blocks any direction completely; its effect is to simply decrease the power in certain directions. In the other limit of $\kappa = \infty$, the antenna has 0 dB gain for $\phi = 0$ exactly and -30 dB for any other direction. Therefore, it is almost omnidirectional, with the exception of a singleton direction whose probability of occurrence is zero. In this limit case, the antenna will, in practice, decrease all the RSS measurements by 30 dB, and that should be the only perceptible effect.

Panel 3.5b shows the result of adding the effects from our virtual antenna with $\kappa = 1$ to the data collected during one of the robot trajectories in the basketball field. The figure compares the empirical distribution of the IM parameters when using the original and modified datasets. We see straightaway three clear effects attributable to the addition of the artificial antenna: the notorious increase and decrease of the averages of σ and z_0 respectively, and an increase of the dispersion of n. Panel 3.5a shows an example of the effect of an antenna with $\kappa = 0.1$ on the RSS-distance scatter plot for a node in the basketball field. The relative variations of n, z_0 and σ in this particular example were in the order of 5%, 12% and 57% respectively.

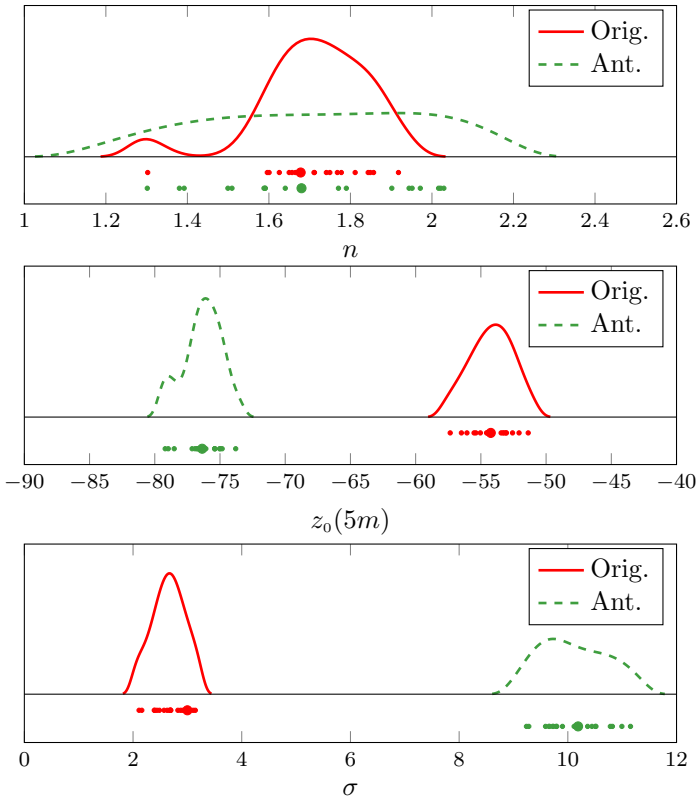
Figure 3.6 shows the effect of a progressive change of the antenna directivity (value of κ) on the empirical distributions of the model parameters in our three experimental environments. The vertical axes show the relative changes of the mean and standard deviation of the model parameters' distributions with respect to their respective values without the effect of the virtual antennas. At first glance, we see that, in general, the changes in the parameters' distribution depend on the type of environment. Panel 3.6a shows that the change on the average path-loss exponent was less than 1% in the basketball field, and up to 6% and 10% in the office and lobby respectively. This suggests that the average path-loss exponent is not strongly affected by the antenna directivity. Its dispersion, however, can be strongly altered (see Panel 3.6b). In the basketball field and lobby, the standard deviation of n can experience variations of more than 100% and 80% respectively, and 25% in the office. This suggests that the antenna directivity has a stronger relative impact on the path-loss exponent in open homogeneous environments. When the environment is homogeneous, the major source of uncertainty is the HW variability. This explains the larger relative importance of the antenna directivity in the basketball field and lobby.

The variation of the average z_0 was as expected given the behaviour of our virtual antenna: a null effect for smaller κ values and a decrease of 30 dB for larger ones (see Panel 3.6c). Note that the figure presents relative changes, and that z_0 is negative. Therefore, a decrease in the relative average implies an increase in the magnitude of the relative average. The dispersion of z_0 can also change with the antenna directivity (see Panel 3.6d). This time the basketball affected the least, with maximum variations of 10%. Variations in the lobby and office were as much as 40% and 20% respectively.

The average and standard deviation of σ increased together with the directivity, as was expected (see panels 3.6e and 3.6f). The strongest relative average increase appeared in the basketball field and lobby, where the relative contribution of the HW variability to the model uncertainty is the highest among our environments.

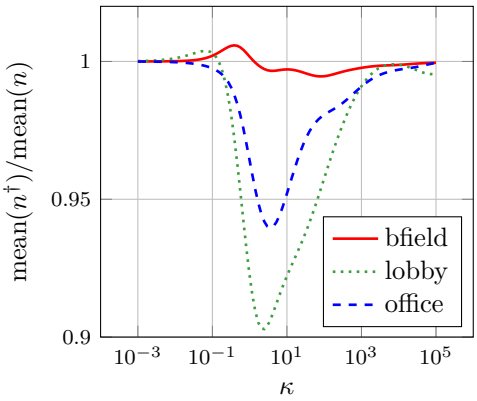


(a) Comparison of RSS-distance scatter plots for a node in the basketball field without and with a directional antenna ($\kappa = 0.1$).

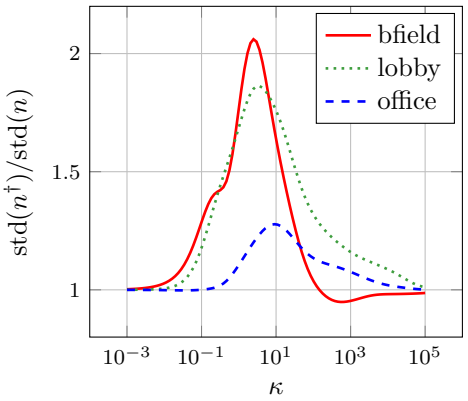


(b) Example of the effect of a directional antenna ($\kappa = 1$) on the parameters of IMs (Orig.=original distribution, Ant.=distribution including the antenna effects). Data collected in one robot trajectory.

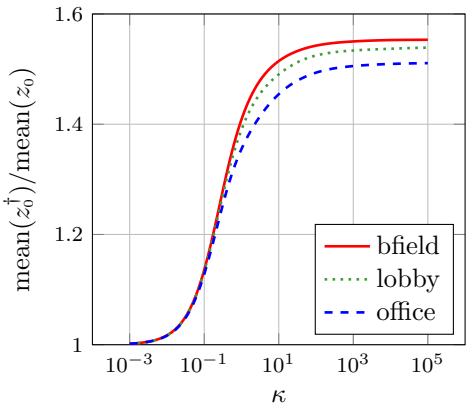
Figure 3.5: Examples of the effect of the antenna directivity in IMs. Data collected in the basketball field



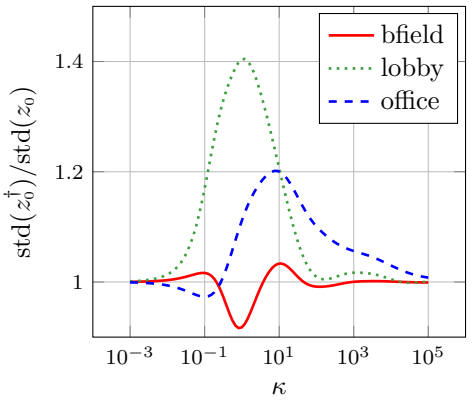
(a) Average n



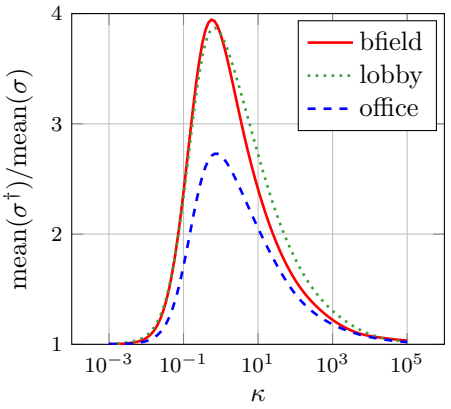
(b) Standard deviation of n



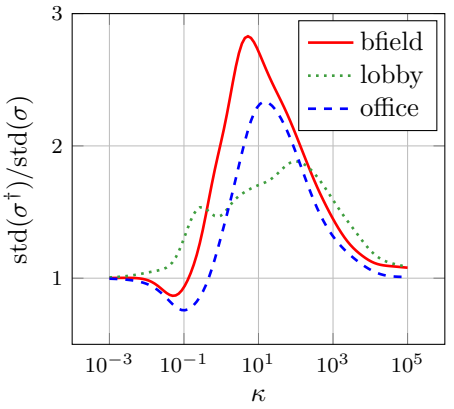
(c) Average z_0



(d) Standard deviation of z_0



(e) Average σ



(f) Standard deviation of σ

Figure 3.6: Relative increments of the mean and standard deviation of the model parameters' empirical distributions as a function of antenna directivity. Parameters with † refer to models with the addition of a virtual antenna, and those without it to the original models.

3.4 Effects of the Environment

The average and dispersion of the distribution of the path-loss exponent were largely affected by the clutteredness and homogeneity of the environment. As expected, on average n was higher in obstructed environments (office) than in more open ones (basketball field and lobby) (see Figure 3.3 and Table 3.1). The dispersion of n can be explained in terms of the different robot-node LoS conditions during the experiments. In cluttered environments, there will on average be more obstacles between the communicating radios as the T-R distance increases. This will curve the trend of the RSS-distance model downwards (see Figure 2.3), an effect which increases the slope of a linear model fit to the observations. Broadly speaking, the higher the number and the greater the severity of obstructions present between a node and the robot, the greater the curvature will be; therefore, n will be larger as well. In this way, varying LoS conditions will induce variability in the path-loss exponent.

Different trajectories might then result in different average values and dispersion of the empirical distribution of n, as shown in Figure 3.3. We can see that, for different trajectories of the robot within the same environment, the densities for n in the lobby are more separated than in the basketball field and office. As argued previously, the inhomogeneity due to the presence of dominant obstacles partially accounts for this difference. The lobby, being a semi-open space, had relatively similar average value of the path-loss exponent to the basketball field with respect to the office. And because it is more inhomogeneous than the basketball field, in general the dispersion was larger. The office is also inhomogeneous, but more cluttered, and thus it exhibited larger values of n on average with a great dispersion. However, the robot moved through the office along rather similar trajectories, as it had to follow corridors and move around in the few free spaces available. Thus, the densities were similar despite the inhomogeneity of the environment.

The impact of the antenna directivity on the path-loss exponent is, in general, clearly less significant than the effect of the environment. As an example, [Dabin et al., 2003] reported path-loss exponent changes of 10% percent when switching from omnidirectional to directional antennas in the same room. Figure 3.5b shows that the largest relative change in n among IMs was in the order of 20%, whereas the relative changes when switching from the basketball field to the office were in the range of between 60 and 80%.

As discussed previously, the distribution of z_0 is most strongly affected by the HW particularities of the nodes. From the middle plot of Figure 3.3, we can see that the strongest effect resulting from a change of environment was a decrease of its average as the environment become cluttered. As argued before, this is partially a consequence of the selection of $d_0 = 5$.

Varying LoS conditions can also affect the dispersion of the reference power z_0. To illustrate this, consider the situation in the lobby, a semi-open space with some large dominant obstacles like stairs (see panels 1.4b and 1.5b). Depending on the relative position of the nodes and the trajectory of the robot, for some nodes there might be a clear LoS with the robot for longer time than for others. The two extreme cases would be that, given the same trajectory of the robot, for one node the stairs are always in between the node and the robot, whereas for the other there is always a clear LoS. Because of the continuous blockage by the stairs, the first case will present a larger negative z_0 than the second. In other non-extreme cases, the situation will fall in between the two above mentioned cases. Thus, varying LoS conditions will also induce changes in z_0. This can be another factor that explains the increase in the dispersion of z_0 in the office compared to the basketball field and lobby (see std. values in Table 3.1)

From the lowest plot in Figure 3.3, we can see that σ takes smaller and less disperse values in the basketball field. This means that the models have a relatively low uncertainty with respect to the models used in the other environments. In the office, σ shows significantly higher and more disperse values. A relatively short dispersion suggests that all the individual models have similar level of uncertainty, whereas a lengthier dispersion indicates that the uncertainty varies a great deal among models. As discussed at the beginning of this section, this can be explained as being the result of having nodes with very different LoS conditions in relation to the robot trajectories. Clearly σ will be larger in deployments with variable LoS conditions.

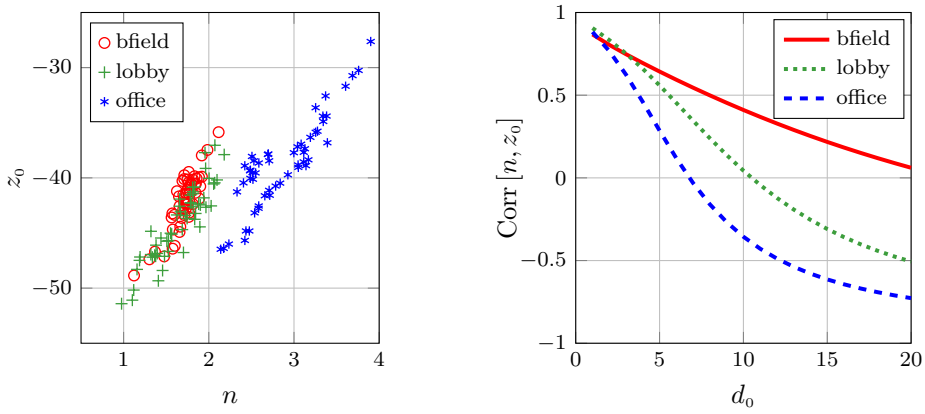
3.5 Correlation between n and z_0

The model parameters can be strongly correlated. Panel 3.7a shows a scatter plot of the values for n and z_0 from the IMs identified from our experimental data, where the correlation is evident. The value of the correlation coefficient depends on the selected value of the reference distance d_0. Table 3.2 presents this value for $d_0 = 1$ and $d_0 = 5$. Panel 3.7b displays its value as a function of d_0 for our three experimental environments.

Table 3.2: Correlation between z_0 and n

d_0	**bfield**	**lobby**	**office**
1	0.87	0.91	0.88
5	0.64	0.56	0.29

The change of sign in the correlation coefficient in the lobby and office is a consequence of the (concave) curvature of the RSS-distance scatter plots observed



(a) z_0-n cloud plot for $d_0 = 1$.

(b) Corr $[n, z_0]$ as a function of d_0.

Figure 3.7: Correlation between z_0 and n.

in those particular environments (see Figure 2.3) and the impossibility of the log-distance model to explain this trend. As argued in Subsection 2.6.2, the limitation of the log-distance model with respect to linearity results in systematically biased RSS predictions. This bias is positive for small distance values and negative for large ones (see Figure 2.3). Additionally, the magnitude of the bias depends on the slope of the linear model fit to the data; that is, it depends upon n. Increments in n will increase the bias for smaller d values, and consequently also the intercept of the linear model along the y axis (z_0 for $d_0 = 1$ m). The same increments in n, on the other hand, will produce a negative bias for large enough distances. This explains the change of sign in the curves of the correlation in Figure 3.7.

As we have just argued, the structural limitation of the log-normal model induces correlation among its parameters. This contributes even more to the interrelation of factors that affect the parameters, and therefore one can no longer assume independence between the factors that affect them.

Knowing the value of the correlation between n and z_0 can be useful, for example, when creating prior distributions for the model parameters.[4] In that case, one might want to select a value of d_0 that eliminates this correlation. However, as shown in Panel 3.7b, this value depends on the type of environment. Interestingly, we can also see how the correlation tends to be similar in all of the environments for smaller d_0 values. Thus, the selection of $d_0 = 1$ seems a reasonable choice, which results in a correlation of around 0.9 in all environments.

[4]This will be needed later in Chapter 7.

3.6 Conclusions

When identifying an empirical log-normal model using training data collected with different nodes, the model parameters can vary significantly depending on the particular nodes being used and their location. This is a direct consequence of the HW and environmental variability. The parameters are affected differently by these factors: HW variability mostly affects the reference power, whereas the environment has a stronger impact in the path-loss exponent. However, they interact in a complex manner that makes it difficult to account accurately for their independent effects.

Although n is indeed mostly affected by the environment (being responsible for between 60% and 80% of the relative change in IMs in one of our experiments), we have demonstrated that in practice it can be significantly affected by the antenna directivity (e.g. responsible for as much as 20% of relative change in one of our tests), and that the extent of this impact depends on the type of environment. While the average n does not change much, the more open and homogeneous the environment is the more its dispersion will increase with the directionality of the antenna. Our experiments also demonstrated that increasing the antenna directionality can strongly increase the average value and dispersion of σ. The reference power, z_0, is directly affected by the directivity, and generally its dispersion increases with it as well.

The variability of the model parameters with respect to HW and environmental factors acquires a special importance in WSNs, and uncovers the potential effects of a) using only one pair of nodes to identify a common model, b) the (often unrealistic) assumption of antenna omnidirectionality and c) the inhomogeneity of the environment. The ultimate question for us is how this variability affects the performance of localization, and whether it is relevant or not. This will be studied in Chapter 5.

Due to the variability of the model parameters, real WSN deployments are better represented by probability distributions of the model parameters rather than deterministic values. From this perspective, the dots shown in Figure 3.3 can be considered *realizations* of the random variables. Furthermore, the model parameters are in general strongly correlated. This implies that the classical assumption of independence between the impact of HW and environmental factors (the first ones affecting only z_0 and the latter only n) is not realistic in WSNs. Due to the intricacy of the effects of these factors on the model parameters, their distribution represents the combined effect of the particular HW being used and the peculiarities of the site of deployment.

Chapter 4

Source Localization

Contents

The previous chapters were fundamentally dedicated to studying in detail the log-normal model as a predictor of the RSS. This chapter is dedicated to the localization algorithms relevant to this thesis. Both well-known and new methods will be presented, and their properties will be analyzed and compared analytically and via simulations. The main focus is on methods that use the log-normal model. However, the algorithms can also be used directly when the measurement model is the additive Gaussian noise (AGN), which is characteristic of ToA-based localization. For completeness, we will present the results for both models.

The first three sections will serve as a reference for the rest of the chapter. Section 4.1 introduces the problem and the basic nomenclature used. Then, Section 4.2 presents a detailed study of the properties of several important distance estimators. The exposition continues in Section 4.3 presenting the Cramér Rao lower bound (CRLB) for the localization problem, which serves as a reference bound against which different estimators can be compared.

Among the possibilities available, LS and ML estimation techniques are selected in this thesis to solve the localization problem. Both methods lead naturally to non-linear problems that require iterative solvers to find the solution. Section 4.4 is dedicated to localization using the LSE criterion. The problem can be formulated in two different ways: a) minimization of the error of the distance estimates and b) when using RSS, also minimization of the measurement errors. These possibilities are analyzed in subsections 4.4.1 and 4.4.2 respectively. Then, in Subsection 4.4.3 the different methods are compared, which will help us to choose the best one in its class for further comparison with other algorithms in later sections.

Although the equations that describe the localization problem are non-linear, there are certain algebraic transformations that lead to a linear problem. Section 4.5 is devoted to a well-known such transformation that has been studied mostly in ToA-based localization. We first examine in subsection 4.5.1 the possibility of using the classic OLS to solve the linearized problem. Then, in subsection 4.5.2 we propose and study two variations based on GLS estimation, both of which increase significantly the performance compared to using OLS. The section then concludes in Subsection 4.5.3 with a comparison of all the linear methods studied using simulations.

Section 4.6 presents a solution to our localization problem using ML estimation, and introduces the related relevant classic theory.

The chapter concludes in Section 4.7 with a comparison among the best methods studied in the chapter and an elaborated discussion about them, which will allow us to decide what methods will be most relevant in the rest of the thesis.

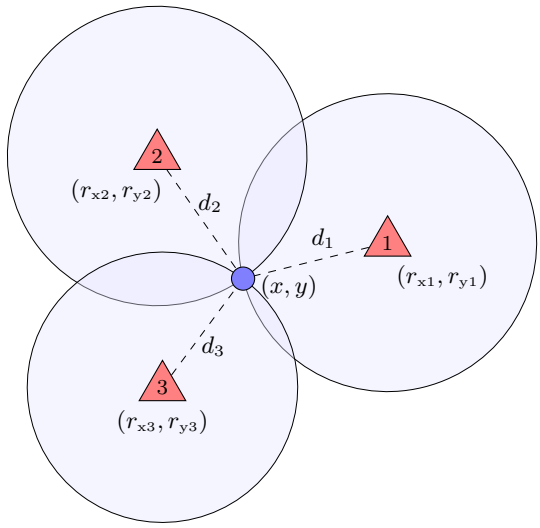


Figure 4.1: Geometry of a basic localization problem when using three beacons.

4.1 Problem Geometry

Let us consider a problem in which we have a set of K beacons with a known location and a node whose position is to be estimated. Without loss of generality and for simplicity's sake, we will consider a localization framework in two dimensions. Let us define,

$\vec{\boldsymbol{s}} = (x, y)^t$	the coordinates of the node,
$\vec{\boldsymbol{r}}_{\mathrm{k}} = (r_{\mathrm{xk}}, r_{\mathrm{yk}})^t$	the coordinates of the k-th beacon, $k = \{1, \ldots, K\}$,
$\boldsymbol{R} = (\vec{\boldsymbol{r}}_1, \ldots, \vec{\boldsymbol{r}}_{\mathrm{K}})^t$	the matrix of coordinates for all the beacons,
$d_{\mathrm{k}} = \|\|\vec{\boldsymbol{s}} - \vec{\boldsymbol{r}}_{\mathrm{k}}\|\|$	the true distance between the k-th beacon and the node,
$\boldsymbol{d} = (d_1, \ldots, d_{\mathrm{K}})^t$	the vector of the K true distances,
$\hat{d}_{\mathrm{k}}$	the distance estimate from the node to the k-th beacon,
$\hat{\boldsymbol{d}} = (\hat{d}_1, \ldots, \hat{d}_{\mathrm{K}})^t$	the vector for the K distance estimates.

Figure 4.1 depicts the geometry of the problem for three beacons. Knowing the true node-beacon distances, the node's position can be calculated as the point at which the three circumferences intersect. For a problem with K beacons, finding the intersection of the K circles implies having to solve the following system of non-linear equations:

$$\begin{aligned}(r_{\mathrm{x}1}-x)^2+(r_{\mathrm{y}1}-y)^2&=d_1^2\\(r_{\mathrm{x}2}-x)^2+(r_{\mathrm{y}2}-y)^2&=d_2^2\\&\cdots\\(r_{\mathrm{x}\mathrm{K}}-x)^2+(r_{\mathrm{y}\mathrm{K}}-y)^2&=d_{\mathrm{K}}^2.\end{aligned}\tag{4.1}$$

In real problems the true distances are unknown, and instead we have available some sort of noisy distance measurements. Due to this noise, the three circumferences in Figure 4.1 will not intersect all at one point, leading to an inconsistent system of equations. In general, then, there is no known exact analytical solution of the system (4.1) given the presence of noisy measurements. Finding a solution then requires other types of methods, some of which we will explore based on LS and ML estimation.

4.2 Distance Estimation

Many localization algorithms rely on the availability of node-beacon distance estimates. Their statistical properties will depend on the properties of the distance estimators that they use, and therefore it is important to know them well.

In this section, we present some important distance estimators for the AGN and log-normal models together with their statistical properties. We will also present estimators for distance powers (i.e. $\widehat{d^m}$) when using the log-normal model, which will be needed later when presenting the LLS position estimators.

4.2.1 Additive Gaussian Noise Model

Let us consider a distance measurement system that gives us the true position corrupted with some noise. Such a model can be written as

$$z = d + \epsilon, \tag{4.2}$$

where z is the measured distance, d is the true distance and ϵ is a random variable characterizing the error. If we have a technology whose measurements can be modelled in this manner, then it is possible to construct an estimator of the distance as $\hat{d} = z$. Thus, the estimator can also be characterized as

$$\hat{d} = d + \epsilon. \tag{4.3}$$

If, on average, the error is zero ($\mathrm{E}[\epsilon] = 0$), then the estimator will give us on average the true distance as an estimate and is said to be unbiased. Typically,

ϵ is assumed to be normally distributed ($\epsilon \sim N(0, \sigma^2)$). In that case, $\hat{d}$ is also normally distributed ($\hat{d} \sim N(d, \sigma^2)$). Moreover, (4.3) is the ML estimator, which is efficient. To prove it, consider the likelihood of a measurement z

$$l(z, d) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\frac{(z-d)^2}{\sigma^2} \right\}.$$

The loglikelihood is

$$L(z, d) = -\ln(\sigma\sqrt{2\pi}) - \frac{1}{2}\frac{(z-d)^2}{\sigma^2}.$$

The Fisher information is then

$$\mathrm{E}\left[\left(\frac{\partial L(z,d)}{\partial d}\right)^2\right] = \mathrm{E}\left[\left(\frac{(z-d)}{\sigma^2}\right)^2\right] = \frac{1}{\sigma^2}\mathrm{E}\left[\left(\frac{z-d}{\sigma}\right)^2\right] = \frac{1}{\sigma^2}\mathrm{E}\left[z_{\mathrm{n}}^2\right],$$

where $z_{\mathrm{n}} = (z-d)/\sigma$. Note that $z_{\mathrm{n}} \sim N(0,1)$, and therefore $z_{\mathrm{n}}^2 \sim \chi^2$ with one degree of freedom, for which $\mathrm{E}\left[z_{\mathrm{n}}^2\right] = 1$. The Fisher information is then $1/\sigma^2$ and the CRLB becomes σ^2. Therefore, the ML distance estimator of (4.3), whose variance is σ^2, is efficient.

4.2.2 RSS-Based Distance Estimation

There are different ways in which one can estimate distance from RSS measurements. One option is to create a distance-RSS mapping from empirical data data, e.g. fitting a function straight from measurements [Yang and Chen, 2009, Caballero et al., 2008a]. Another possibility is to use kernel methods to find a direct mapping from the RSS space to a function of the node-beacon distances, after which the distance can be recovered using the inverse of this function.[1] Perhaps the most common way is to find an estimator based on an RSS-distance model. This is the method that we will follow, in particular using the log-normal model.

Consider the log-normal model

$$z = \bar{z} + \epsilon, \tag{4.4}$$

where

$$\bar{z} = z_0 - 10\, n \log(d/d_0) \tag{4.5}$$
$$\epsilon \sim N(0, \sigma^2).$$

[1] For example, in [Lim et al., 2006] the log-distances are calculated as a linear combination of the type $\log(\boldsymbol{d}) = \sum_{i=1}^{K} z_i \boldsymbol{u}_i$, where the logarithm is element-wise, and z_i and $\boldsymbol{u}_i$ are respectively the RSS measurement and a vector associated with the i-th beacon. The idea is then to find the vectors $\boldsymbol{u}_i$ in the calibration step, chosen so that the estimated distance errors are minimized.

We shall now calculate the CRLB for distance estimators using the previous model, which will serve as a reference to compare the estimators that will be reviewed later.

The likelihood of one measurement z is

$$l(z,d) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ -\frac{1}{2}\frac{(z-\bar{z})^2}{\sigma^2} \right\}, \tag{4.6}$$

and the log-likelihood

$$L(z,d) = -\ln(\sigma\sqrt{2\pi}) - \frac{1}{2}\frac{(z-\bar{z})^2}{\sigma^2}. \tag{4.7}$$

The Fisher information then becomes

$$\mathrm{E}\left[\left(\frac{\partial L(z,d)}{\partial d}\right)^2\right] = \mathrm{E}\left[\left(\frac{z-\bar{z}}{\sigma}\right)^2\left(\frac{10n}{d\sigma\ln 10}\right)^2\right] = \left(\frac{10n}{d\sigma\ln 10}\right)^2 \mathrm{E}\left[z_{\mathrm{n}}^2\right],$$

where $z_{\mathrm{n}} = \left(\frac{z-\bar{z}}{\sigma}\right) \sim N(0,1)$. Thus, $z_{\mathrm{n}}^2 \sim \chi^2$ with one degree of freedom, for which $\mathrm{E}\left[z_{\mathrm{n}}^2\right] = 1$. Therefore, the CRLB is

$$\mathrm{CRLB} = \left(\frac{\sigma\ln 10}{10n}\right)^2 d^2 = \gamma^2\, d^2, \tag{4.8}$$

where

$$\gamma \equiv \frac{\sigma\ln 10}{10n}. \tag{4.9}$$

As we can see, the minimum possible variance of any unbiased estimator increases linearly with the variance of the noise and quadratically with the distance to be estimated. Also, the greater the path-loss exponent the smaller the minimum variance will be. As discussed in Chapter 2, higher path loss exponents can be found in cluttered environments. However, these environments tend to also be more inhomogeneous, leading to an increase of σ^2. Depending on the particular environment, the aggregated effect can then theoretically be either a decrease or an increase in the CRLB.

The **ML distance estimator** is the value of d that maximizes (4.7), which clearly should satisfy $z = \bar{z}$. Applying this to (4.5) and solving for d, we obtain the ML estimator

$$\hat{d}_{\mathrm{ML}} = d_0\, 10^{\frac{z_0 - z}{10\,n}}. \tag{4.10}$$

Substituting (4.4) for z in (4.10) and rearranging terms, we can write

$$\hat{d}_{\mathrm{ML}} = d\, 10^{\frac{\epsilon}{10\,n}} = d\exp\left\{\frac{\ln 10}{10n}\epsilon\right\} = d\, e^{\epsilon'},\ \ \epsilon' \sim N(0,\gamma^2). \tag{4.11}$$

The variable $e^{\epsilon'}$ is log-normally distributed, with the following mean and variance

$$\mathrm{E}\left[e^{\epsilon'}\right] = e^{\gamma^2/2} \tag{4.12}$$

$$\mathrm{Var}\left[e^{\epsilon'}\right] = (e^{\gamma^2} - 1)e^{\gamma^2}. \tag{4.13}$$

Therefore, we can see that the estimator is log-normally distributed. Let us define

$$\Gamma \equiv e^{\gamma^2/2}. \tag{4.14}$$

Using the previous definition together with (4.11), (4.12) and (4.13), we can readily see that the mean and variance of the ML estimator are, respectively,

$$\mathrm{E}\left[\hat{d}_{\mathrm{ML}}\right] = d\,\Gamma \tag{4.15}$$

$$\mathrm{Var}\left[\hat{d}_{\mathrm{ML}}\right] = d^2(\Gamma^2 - 1)\Gamma^2. \tag{4.16}$$

Hence, the ML distance estimator is biased. Noting that $\Gamma \geq 1$, [2] we can see that the bias is always positive, and therefore the estimator will give on average larger values than the true distance. Its mean square error (MSE) is

$$\mathrm{MSE}\left[\hat{d}_{\mathrm{ML}}\right] = \mathrm{Var}\left[\hat{d}_{\mathrm{ML}}\right] + \left(\mathrm{Bias}\left[\hat{d}_{\mathrm{ML}}\right]\right)^2 = d^2(\Gamma^4 - 2\Gamma + 1). \tag{4.17}$$

As we can see from 4.14 and 4.9, the MSE grows exponentially with σ^2.

Based on (4.15), we can construct the following **unbiased distance estimator** [Patwari et al., 2007]:

$$\hat{d}_{\mathrm{U}} = \frac{\hat{d}_{\mathrm{ML}}}{\Gamma} = \frac{d_0}{\Gamma}\, 10^{\frac{z_0 - z}{10\, n}}, \tag{4.18}$$

which, when using equation (4.11), can be equivalently written as

$$\hat{d}_{\mathrm{U}} = \frac{d}{\Gamma}\, e^{\epsilon'}, \quad \epsilon' \sim N(0, \gamma^2). \tag{4.19}$$

Its variance is

$$\mathrm{Var}\left[\hat{d}_{\mathrm{U}}\right] = \mathrm{Var}\left[\frac{\hat{d}_{\mathrm{ML}}}{\Gamma}\right] = \frac{1}{\Gamma^2}\mathrm{Var}\left[\hat{d}_{\mathrm{ML}}\right] = d^2(\Gamma^2 - 1). \tag{4.20}$$

[2] The exponent in the definition (4.14) is non-negative, which implies that $\Gamma \geq 1$. The equality $\Gamma = 1$ requires that $\gamma = 0$, which in turn requires that either $\sigma = 0$ or $n = \infty$, which are degenerate cases.

When comparing the variances of the ML and unbiased estimators, we can see that the former is larger than the latter by a factor of Γ^2. Therefore, the variance of the unbiased distance estimator is always smaller than the variance of the ML estimator. Additionally, being unbiased, the MSE of the unbiased estimator is equal to its variance, and therefore the MSE of the unbiased estimator is always smaller than that of the ML.

When comparing the ML and unbiased estimators with the CRLB, we can see that their MSE grows exponentially with σ^2, as opposed to linearly in the CRLB. A natural question that arises then is whether we can find better estimators than the ones we have seen so far, and ultimately whether an efficient estimator exists. Using again the results from [Chitte et al., 2009a], it can be demonstrated that there is no efficient distance estimator under (4.4), which means that no unbiased estimator attains the CRLB. Furthermore, Chitte et al. demonstrated that (4.18) is the best unbiased estimator in the MSE sense. As an alternative, they proposed a linear minimum mean square error (LMMSE) estimator, wherein the linearity is with respect to the received power measured in W instead of dBm. Following the formulation of the log-normal model used in this thesis, [3] we can reach the same result as follows. Let us rearrange (4.4) as

$$\underbrace{d_0\, 10^{\frac{z_0 - z}{10\, n}}}_{z_\mathrm{w}} = d\, 10^{-\varepsilon/10n}, \tag{4.21}$$

where z_w is the observation. The linear estimator that we are looking for has the form $\hat{d}_v = \kappa\, z_\mathrm{w}$, and the MSE to be minimized is $\mathrm{E}\left[(d - \kappa\, z_\mathrm{w})^2\right]$. The value of κ that minimizes the MSE is $\kappa = \Gamma^{-3}$ (see Appendix A.1). The proposed **linear MMSE estimator** is then

$$\hat{d}_\mathrm{V} = \Gamma^{-3} d_0\, 10^{\frac{z_0 - z}{10\, n}} = \frac{\hat{d}_\mathrm{ML}}{\Gamma^3}. \tag{4.22}$$

From (4.15) and (4.16), its expected value, variance and MSE are:

$$\mathrm{E}\left[\hat{d}_\mathrm{V}\right] = \frac{d}{\Gamma^2} \tag{4.23}$$

$$\mathrm{Var}\left[\hat{d}_\mathrm{V}\right] = d^2 \frac{\Gamma^2 - 1}{\Gamma^4} \tag{4.24}$$

$$\mathrm{MSE}\left[\hat{d}_\mathrm{V}\right] = d^2 (1 - \Gamma^{-2}) \tag{4.25}$$

In contrast to the ML distance estimator, the bias of the LMMSE estimator is always negative, and therefore the predicted distances are, on average, smaller

[3] The formulation used by Chitte *et al.*. uses natural logarithms and different definitions for the model parameters.

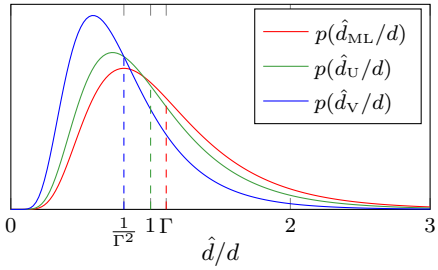


Figure 4.2: Distribution of the normalized distance estimators $\hat{d}/d$ for $n = 2$ and $\sigma = 4$. The dashed lines indicate the respective expected distance estimates.

than the true ones. Also, because $\Gamma \geq 1$, the MSE is bounded by d^2. Compare this with the exponential growth of the MSE in the ML and unbiased estimators. In fact, because $\gamma^2 > (1 - \Gamma^{-2})\ \forall \gamma > 0$, we see that the MSE is always smaller than the CRLB (see (4.8)).

Figure (4.2) shows a comparison of the distribution of the three different distance estimators revised so far, in which we can clearly see the positive, zero and negative bias of the ML, unbiased and LMMSE estimators, respectively.

4.2.3 RSS-Based Estimation of d^m

In the following sections, it will prove useful to find estimators of distance powers. For convenience, let us define $P \equiv d^m$. In principle, one could think of using the m-th power of the estimators of d as the estimator of P. In the case of the ML distance estimator, we would have

$$\hat{P}_{\mathrm{ML}} \equiv \hat{d}^m_{\mathrm{ML}}, \tag{4.26}$$

which in fact is the **ML estimator** of d^m [Chitte et al., 2009a]. In order to calculate its statistical properties, we start from (4.11), from which we have $\hat{d}^m_{\mathrm{ML}} = d^m e^{m\epsilon'}$ with $m\epsilon' \sim \mathcal{N}(0, (m\gamma)^2)$. Let us now define

$$\gamma_m \equiv m\gamma = \frac{m\sigma \ln 10}{10n} \tag{4.27}$$

$$\Gamma_m \equiv e^{\gamma_m^2/2}, \tag{4.28}$$

and so $m\epsilon' \sim \mathcal{N}(0, \gamma_m^2)$. Thus, the expectation and variance of $e^{m\epsilon'}$ can be calculated from (4.12) and (4.13), respectively, simply by exchanging γ for γ_m. Using this equivalence in (4.15), (4.16) and (4.17) and the definition (4.28), the

mean, covariance and MSE of (4.26) can be expressed as:

$$\mathrm{E}\left[\hat{P}_{\mathrm{ML}}\right] = d^m \Gamma_m \tag{4.29}$$

$$\mathrm{Var}\left[\hat{P}_{\mathrm{ML}}\right] = d^{2m}(\Gamma_m^2 - 1)\Gamma_m^2 \tag{4.30}$$

$$\mathrm{MSE}\left[\hat{d}_{\mathrm{ML}}\right] = d^{2m}(\Gamma_m^4 - 2\Gamma_m + 1). \tag{4.31}$$

Clearly, the estimator is biased. Chitte et al. also demonstrated that the best **unbiased estimator** of d^m is

$$\hat{P}_{\mathrm{U}} \equiv \frac{\hat{d}_{\mathrm{ML}}^m}{\Gamma_m} = \frac{\hat{P}_{\mathrm{ML}}}{\Gamma_m}. \tag{4.32}$$

Using (4.30), its variance (and MSE) can be readily calculated as:

$$\mathrm{Var}\left[\hat{P}_{\mathrm{U}}\right] = d^{2m}(\Gamma_m^2 - 1). \tag{4.33}$$

Compare now the definitions for Γ in (4.9) and for Γ_m in (4.28), and note that

$$\Gamma_m^l = \exp\left\{\frac{l}{2}\left(\frac{m\sigma \ln 10}{10n}\right)^2\right\} = \exp\left\{\frac{m^2 l}{2}\left(\frac{\sigma \ln 10}{10n}\right)^2\right\} = \Gamma_1^{m^2 l} = \Gamma^{m^2 l}. \tag{4.34}$$

Thus, $\Gamma_m \neq \Gamma^m = \Gamma_1^m$, and therefore $\hat{P}_{\mathrm{U}} \neq (\hat{d}_{\mathrm{U}})^m$. In other words, unlike in the case of the ML estimator, the unbiased estimator of d^m is not the m-th power of the unbiased estimator of d.

Chitte et al. additionally proposed a **LMMSE estimator** of d^m. Using the formulation provided in this thesis, we can obtain the same results using (4.21) and letting the estimator take the form $\hat{P}_{\mathrm{V}} = \kappa\, z_{\mathrm{w}}^m$. The value of κ that minimizes the MSE $\mathrm{E}\left[(d^m - \kappa\, z_{\mathrm{w}}^m)^2\right]$ is $\kappa = \Gamma_m^{-3}$ (see Appendix A.1), which leads to the following LMMSE estimator of d^m:

$$\hat{P}_{\mathrm{V}} \equiv \frac{\hat{d}_{\mathrm{ML}}^m}{\Gamma_m^3} = \frac{\hat{P}_{\mathrm{ML}}}{\Gamma_m^3}. \tag{4.35}$$

Using (4.29) and (4.30), its expectation, variance and MSE can be readily calculated, resulting in

$$\mathrm{E}\left[\hat{P}_{\mathrm{V}}\right] = d^m \Gamma_m^{-2} \tag{4.36}$$

$$\mathrm{Var}\left[\hat{P}_{\mathrm{V}}\right] = d^{2m}\frac{\Gamma_m^2 - 1}{\Gamma_m^4} \tag{4.37}$$

$$\mathrm{MSE}\left[\hat{P}_{\mathrm{V}}\right] = d^{2m}(1 - \Gamma_m^{-2}). \tag{4.38}$$

The MSE of the LMMSE estimator is now bounded by d^{2m}, and again it does not grow exponentially with σ^2.

The fundamental problem with this estimator is that it can have a strong downward bias (guaranteed by the fact that $\Gamma \geq 1$). Effectively, from (4.36) and using (4.34), we can write

$$\mathrm{E}\left[\hat{P}_{\mathrm{V}}\right] = \frac{d^m}{\Gamma_m^2} = \frac{d^m}{\Gamma^{2m^2}} \implies \left(\mathrm{E}\left[\hat{P}_{\mathrm{V}}\right]\right)^{1/m} = \frac{d}{\Gamma^{2m}}, \tag{4.39}$$

which clearly shows a bias that increases rapidly with m. Table 4.1 presents the values of the relative bias for different values of m for an sample situation with $\Gamma = 1.1$ (e.g. $n = 2$ and $\sigma = 3.8$). As we can see, already when $m = 2$ the

Table 4.1: Relative error of LMMSE distance estimator ($\Gamma = 1$).

m	1	2	4
$\frac{1}{\Gamma^{2m^2}}$	0.83	0.47	0.05
$\frac{1}{\Gamma^{2m}}$	0.83	0.68	0.47

estimator gives, on average, 47% of the true value d^m, which represents 68% of d. For $m = 4$, on average the estimator will give the 5% of the true fourth power of the distance, representing 47% of d.

4.3 CRLB Localization Performance Bounds

The CRLB is a fundamental limit that states what is the minimum variance achievable by any unbiased estimator given a complete probabilistic measurement model. It therefore does not depend on the estimation method. Because of these properties, it is typically used to assess the goodness of different estimation algorithms. In this section, we present the CRLB for localization problems when using the additive Gaussian and log-normal measurement models. These results will then be used to compare the performance of the algorithms presented later.

In both cases the measurements are normally distributed. For a generic model with observations $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{C}(\boldsymbol{\theta}))$, the Fisher information matrix (FIM) can be expressed as [Kay, 1993, p.47]:

$$\begin{aligned}\left[\mathbf{FIM}(\boldsymbol{\theta})\right]_{ij} = &\left[\frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_i}\right]^t \boldsymbol{C}^{-1}(\boldsymbol{\theta}) \left[\frac{\partial \boldsymbol{\mu}(\boldsymbol{\theta})}{\partial \theta_j}\right] \\ &+ \frac{1}{2}\,\mathrm{Tr}\left[\boldsymbol{C}^{-1}(\boldsymbol{\theta})\frac{\partial \boldsymbol{C}(\boldsymbol{\theta})}{\partial \theta_i}\boldsymbol{C}^{-1}(\boldsymbol{\theta})\frac{\partial \boldsymbol{C}(\boldsymbol{\theta})}{\partial \theta_j}\right].\end{aligned} \tag{4.40}$$

As we can see, the FIM can be divided in two parts, one associated with the information available in the mean $\boldsymbol{\mu}$ (first line) and another with the information in the covariance matrix $\boldsymbol{C}$ (second line). For both models discussed in this chapter, we consider IID measurements that result in constant covariance matrices. Thus, the model for the covariance does not provide any information, and the second line of (4.40) will be zero.

CRLB for Gaussian Additive Noise Model

Applying (4.40) to the AGN model (4.2) leads to

$$\mathbf{FIM}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \left[\frac{\partial \boldsymbol{d}}{\partial \vec{\boldsymbol{s}}}\right]^t \boldsymbol{\Sigma}^{-1} \left[\frac{\partial \boldsymbol{d}}{\partial \vec{\boldsymbol{s}}}\right],$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the measurements, and the Jacobian is

$$\frac{\partial \boldsymbol{d}}{\partial \vec{\boldsymbol{s}}} = \begin{bmatrix} \frac{x-r_{\mathrm{x1}}}{d_1} & \frac{y-r_{\mathrm{y1}}}{d_1} \\ \cdots & \cdots \\ \frac{x-r_{\mathrm{xK}}}{d_{\mathrm{K}}} & \frac{y-r_{\mathrm{yK}}}{d_{\mathrm{K}}} \end{bmatrix} = \boldsymbol{D}^{-1}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}),$$

where

$$\mathbf{1} = (1, \ldots, 1)^t, \;\; \dim(\mathbf{1}) = K \times 1 \tag{4.41}$$

$$\boldsymbol{R} = (\vec{\boldsymbol{r}}_1, \ldots, \vec{\boldsymbol{r}}_{\mathrm{K}})^t, \;\; \dim(\boldsymbol{R}) = K \times 2 \tag{4.42}$$

$$\boldsymbol{D} = \mathrm{diag}(d_1, \ldots, d_{\mathrm{K}}), \;\; \dim(\boldsymbol{D}) = K \times K \tag{4.43}$$

With this expression, the FIM can be written as:

$$\begin{aligned} \mathbf{FIM}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) &= [\boldsymbol{D}^{-1}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})]^t \boldsymbol{\Sigma}^{-1} \boldsymbol{D}^{-1}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}) \\ &= (\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})^t \boldsymbol{D}^{-1} \boldsymbol{\Sigma}^{-1} \boldsymbol{D}^{-1}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}). \end{aligned}$$

For cases in which $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, the previous expression simplifies to

$$\mathbf{FIM}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \frac{1}{\sigma^2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})^t \boldsymbol{D}^{-2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}).$$

The CRLB then becomes

$$\mathbf{CRLB}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \sigma^2 \underbrace{[(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})^t \boldsymbol{D}^{-2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})]^{-1}}_{\boldsymbol{\Phi}_{\mathbf{d}}} = \sigma^2 \boldsymbol{\Phi}_{\mathbf{d}}(\vec{\boldsymbol{s}}), \tag{4.44}$$

where $\boldsymbol{\Phi}_{\mathbf{d}}(\vec{\boldsymbol{s}})$ depends exclusively on the problem geometry, and for a given set of beacon positions, it only depends on the position to be estimated, $\vec{\boldsymbol{s}}$.

CRLB for the Log-Normal Model

Applying (4.40) to the model (4.4) results in

$$\mathbf{FIM}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \left[\frac{\partial \bar{\boldsymbol{z}}}{\partial \vec{\boldsymbol{s}}}\right]^t \boldsymbol{\Sigma}^{-1} \left[\frac{\partial \bar{\boldsymbol{z}}}{\partial \vec{\boldsymbol{s}}}\right],$$

where $\bar{\boldsymbol{z}}$ is the vector of average RSS. When considering individual models (i.e. each beacon has its own model), the Jacobian is:

$$\frac{\partial \bar{\boldsymbol{z}}}{\partial \vec{\boldsymbol{s}}} = -\frac{10}{\ln 10} \begin{bmatrix} n_1 \frac{x - r_{\mathrm{x}1}}{d_1^2} & n_1 \frac{y - r_{\mathrm{y}1}}{d_1^2} \\ \cdots & \\ n_{\mathrm{K}} \frac{x - r_{\mathrm{x}\mathrm{K}}}{d_{\mathrm{K}}^2} & n_{\mathrm{K}} \frac{y - r_{\mathrm{y}\mathrm{K}}}{d_{\mathrm{K}}^2} \end{bmatrix} = -\frac{10}{\ln 10} \boldsymbol{N}\, \boldsymbol{D}^{-2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}),$$

where $\boldsymbol{N} = \mathrm{diag}(n_1, \ldots, n_{\mathrm{K}})$ and n_{k} is the path-loss exponent of the k-th node's model. With this expression, the FIM can be written as

$$\begin{aligned} \mathbf{FIM}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) &= \left(\frac{10}{\ln 10}\right)^2 [\boldsymbol{N}\boldsymbol{D}^{-2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})]^t \boldsymbol{\Sigma}^{-1} \boldsymbol{N}\boldsymbol{D}^{-2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}) \\ &= \left(\frac{10}{\ln 10}\right)^2 (\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})^t \boldsymbol{D}^{-2}\, \boldsymbol{N}\boldsymbol{\Sigma}^{-1}\boldsymbol{N}\, \boldsymbol{D}^{-2}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}). \end{aligned}$$

For cases in which $\boldsymbol{N} = n\mathbf{I}$ (e.g. when we use a common model) and $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$, the previous expression simplifies to

$$\mathbf{FIM}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \left(\frac{10n}{\sigma \ln 10}\right)^2 (\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})^t \boldsymbol{D}^{-4}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R}).$$

Using the definition (4.9), the CRLB becomes

$$\mathbf{CRLB}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \gamma^2 \underbrace{[(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})^t \boldsymbol{D}^{-4}(\mathbf{1}\vec{\boldsymbol{s}}^t - \boldsymbol{R})]^{-1}}_{\boldsymbol{\Phi}_{\mathbf{p}}(\vec{\boldsymbol{s}})} = \gamma^2 \boldsymbol{\Phi}_{\mathbf{p}}(\vec{\boldsymbol{s}}). \tag{4.45}$$

This expression has two factors: γ^2, which depends on the model parameters, and $\boldsymbol{\Phi}_{\mathbf{p}}(\vec{\boldsymbol{s}})$, which depends on the geometry of the problem. As we can see, when using the log-normal model for localization, the covariance of the best possible unbiased position estimator increases linearly with γ^2, and therefore it increases linearly with σ^2 and decreases proportionally to the inverse of the square path-loss exponent.

4.4 Nonlinear LS Methods

Consider a generic estimation problem in which we have noisy measurements that are a function of the variables that we want to estimate. Let us for the moment define a general measurement model:

$$z = f(\boldsymbol{\theta}) + \epsilon, \tag{4.46}$$

where z is the quantity that we can measure, f is a deterministic function of the parameters, $\boldsymbol{\theta}$, to be estimated and ϵ is a random variable characterizing the measurement error. For a set of K measurements, we can define the vector function $\boldsymbol{f}(\boldsymbol{\theta})$ as $\boldsymbol{f}(\boldsymbol{\theta}) = (f_1(\boldsymbol{\theta}), \ldots, f_K(\boldsymbol{\theta}))^t$. The measurement model is then

$$\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}, \tag{4.47}$$

where $\boldsymbol{z}$ is the vector of observations and $\boldsymbol{\epsilon}$ the (random) vector of errors. In a particular experimental setup, we assume that the parameter $\boldsymbol{\theta}$ receives a deterministic fixed value, that we refer to as the true parameter value, $\boldsymbol{\theta}_t$; this value is used to generate the data. The generative model of that particular setup becomes

$$\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{\theta}_t) + \boldsymbol{\epsilon}. \tag{4.48}$$

Having defined our model, our aim is to find an estimate of $\boldsymbol{\theta}_t$. In order to do so, we pose the problem as a minimization of a cost function that penalizes the difference between the observations and $\boldsymbol{f}(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is an independent variable. A convenient cost function is

$$Q = \boldsymbol{\epsilon}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon} = \left[\boldsymbol{z} - \boldsymbol{f}(\boldsymbol{\theta})\right]^t \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{z} - \boldsymbol{f}(\boldsymbol{\theta})\right], \tag{4.49}$$

which is a quadratic function of the errors leading to an LS problem. The matrix $\boldsymbol{\Sigma}$ is the covariance of the error terms. The estimator of $\boldsymbol{\theta}_t$ is then the value of $\boldsymbol{\theta}$ that minimizes Q, that is,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}}(Q). \tag{4.50}$$

Should function $\boldsymbol{f}$ be linear in $\boldsymbol{\theta}$, say $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{\theta}) = \boldsymbol{T}\boldsymbol{\theta}$, then the solution is given by the GLS estimator [Davidson and MacKinnon, 2004, p. 258]:

$$\hat{\boldsymbol{\theta}}_{\text{GLS}} = (\boldsymbol{T}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{T})^{-1} \boldsymbol{T}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{z}. \tag{4.51}$$

If in addition the errors are uncorrelated and homoskedastic, the covariance matrix, $\boldsymbol{\Sigma}$, can be written as $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, and (4.51) simplifies to

$$\hat{\boldsymbol{\theta}}_{\text{OLS}} = (\boldsymbol{T}^t \boldsymbol{T})^{-1} \boldsymbol{T}^t \boldsymbol{z}, \tag{4.52}$$

which is the OLS estimator [Davidson and MacKinnon, 2004, p. 33].

So far, we have made no assumptions whatsoever about the distribution of the errors, nor have we made any claims about the properties of the estimators other than that the previous solutions minimize the cost function (4.49). If in addition to $\boldsymbol{f}$ being linear we can assume that the errors are zero mean ($\mathrm{E}[\boldsymbol{\epsilon}] = 0$), substituting $\boldsymbol{z} = \boldsymbol{T}\boldsymbol{\theta} + \boldsymbol{\epsilon}$ for $\boldsymbol{z}$ in (4.51) and (4.52) and taking expectation we can readily see that the GLS and OLS estimators will be unbiased. Clearly, otherwise the estimators will be biased. In cases with zero mean errors, it can also be seen that the GLS estimator is the best linear unbiased estimator (BLUE) [Amemiya, 1985, sec 6.1]. If the errors are uncorrelated and homoskedastic, then $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$ and the OLS is then the BLUE. Otherwise, the OLS is inefficient.Yet, if we further assume that the errors are normally distributed, then the GLS estimator is said to be efficient, as it attains the CRLB. Likewise, the OLS is efficient in cases with uncorrelated and homoskedastic errors.

In the most general case in which $\boldsymbol{f}$ is not linear in $\boldsymbol{\theta}$, we have a generalized non-linear least squares (GNLS) problem, which must be solved using iterative numerical methods. A classical approach then is to approximate $\boldsymbol{f}$ with a linear function of $\boldsymbol{\theta}$ and substitute it in (4.49) for its approximation. This requires the availability of an initial guess regarding the parameter $\hat{\boldsymbol{\theta}}_n$, around which we will linearize $\boldsymbol{f}$. The strategy is then to find a succession of vectors, $\boldsymbol{\delta}_n$, that, when added to $\hat{\boldsymbol{\theta}}_n$, result in a successively smaller cost function Q.

In the position estimation problem, the parameters that we want to estimate are the coordinates of the node ($\boldsymbol{\theta} = \vec{\boldsymbol{s}}$), and the observations $\boldsymbol{z}$ can be any function dependent on the position of the node. In a particular setup in which the position to be estimated is fixed to $\vec{\boldsymbol{s}}_{\mathrm{t}}$, the model for the random observations (4.48) becomes

$$\boldsymbol{z} = \boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) + \boldsymbol{\epsilon}, \tag{4.53}$$

and the cost function (4.49)

$$Q = \boldsymbol{\epsilon}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\epsilon} = \left[\boldsymbol{z} - \boldsymbol{f}(\vec{\boldsymbol{s}})\right]^t \boldsymbol{\Sigma}^{-1}\left[\boldsymbol{z} - \boldsymbol{f}(\vec{\boldsymbol{s}})\right]. \tag{4.54}$$

In general, $\boldsymbol{f}(\vec{\boldsymbol{s}})$ is non-linear in $\vec{\boldsymbol{s}}$. If we have an initial guess at the position of the node $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$, we can expand $\boldsymbol{f}(\vec{\boldsymbol{s}})$ in Taylor's series with respect to $\vec{\boldsymbol{s}}$ about $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$ and use this to find a refined position estimate, $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}}$. It can be shown that the new position estimate is (see Appendix A.2 for a detailed derivation):

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}} = \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} + (\mathbf{J}_{\mathrm{n}}^t\boldsymbol{\Sigma}^{-1}\mathbf{J}_{\mathrm{n}})^{-1}\mathbf{J}_{\mathrm{n}}^t\boldsymbol{\Sigma}^{-1}[\boldsymbol{z} - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})], \tag{4.55}$$

where $\mathbf{J}_{\mathrm{n}}$ is a $K \times 2$ Jacobian matrix evaluated at each iteration and defined as

$$\mathbf{J}_{\mathrm{n}} \equiv \left.\frac{\partial \boldsymbol{f}(\vec{\boldsymbol{s}})}{\partial \vec{\boldsymbol{s}}}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} = \begin{pmatrix} \left.\frac{\partial f_1}{\partial x}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \left.\frac{\partial f_1}{\partial y}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \\ \cdots & \cdots \\ \left.\frac{\partial f_K}{\partial x}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \left.\frac{\partial f_K}{\partial y}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \end{pmatrix}. \tag{4.56}$$

Once $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}$ has been calculated, we can use it as a new guess at the position estimate and iterate the process until convergence. This procedure is essentially the classic Newton-Raphson optimization algorithm [Kay, 1993, Sec. 7.7]. As with any local search method, there are no guarantees that it will converge or find the global minimum. Several criteria can be used to detect convergence and stop the iteration [Davidson and MacKinnon, 2004, p. 232]. For example, we could stop it when $\|\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}\|_2 \approx 0$.

Assuming that the iteration converges, then for a large enough n we have $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1} \approx \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$. A useful alternative expression of (4.55) with which we can study the statistical properties of the estimator is (see [Torrieri, 1984], or appendix A.2 for a detailed derivation):

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1} = \vec{\boldsymbol{s}}_{\mathrm{t}} + (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{e}_{\mathrm{L}} + \boldsymbol{\epsilon}) \tag{4.57}$$

$$\boldsymbol{e}_{\mathrm{L}} = [\boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})] - \mathbf{J}_{\mathrm{n}}(\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}). \tag{4.58}$$

Note that the expression includes the true position $\vec{\boldsymbol{s}}_{\mathrm{t}}$. The term $\boldsymbol{e}_{\mathrm{L}}$ is the linearization error, and thus (4.57) shows clearly that the error in the estimator depends on the linearization and measurement errors. Using the previous expression, the bias and covariance of the estimator can be readily calculated as:

$$\mathrm{E}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] - \vec{\boldsymbol{s}}_{\mathrm{t}} = (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{e}_{\mathrm{L}} + \mathrm{E}[\boldsymbol{\epsilon}]) \tag{4.59}$$

$$\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] = [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1}. \tag{4.60}$$

As we can see, due to the linearization error, the estimator will be in general biased even if $E[\boldsymbol{\epsilon}] = 0$. Should $\boldsymbol{f}(\vec{\boldsymbol{s}})$ be linear, then the term $\boldsymbol{e}_{\mathrm{L}}$ vanishes. If in addition $E[\boldsymbol{\epsilon}] = 0$, then the resulting estimator will be unbiased. In the general case of $\boldsymbol{f}(\vec{\boldsymbol{s}})$ not being linear, the more non-linear the function is the more bias it will produce in the estimator.

Using the Taylor-series linearization to solve localization problems was proposed initially by Foy [Foy, 1976], who applied the method to cases in which the measurements available are line of positions (LoPs), relative angles (classical triangulation) and relative distances (multilateration). He also presented the results for TDoA (hyperbolic location), differential angle of arrival (DAoA) and for cases in which different types of measurements are combined. Torrieri applied later the method to hyperbolic localization and AoA problems studying in detail the circular error probability (CEP) and GDoP [Torrieri, 1984] .

4.4.1 Localization Using Range Measurements

When using range measurements for localization, the measurements are the distance estimates from the beacons to the node ($\boldsymbol{z} = \hat{\boldsymbol{d}}$), and the function $\boldsymbol{f}(\vec{\boldsymbol{s}})$ becomes

$$\boldsymbol{d}_{\boldsymbol{f}}(\vec{\boldsymbol{s}}) \equiv \boldsymbol{f}(\vec{\boldsymbol{s}}) = \begin{pmatrix} \|\vec{\boldsymbol{r}}_1 - \vec{\boldsymbol{s}}\|_2 \\ \vdots \\ \|\vec{\boldsymbol{r}}_{\mathrm{K}} - \vec{\boldsymbol{s}}\|_2 \end{pmatrix}, \tag{4.61}$$

where $\|\vec{\boldsymbol{r}}_{\mathrm{k}} - \vec{\boldsymbol{s}}\|_2 = \sqrt{(r_{\mathrm{xk}} - x)^2 + (r_{\mathrm{yk}} - y)^2}$. Then, (4.53) becomes

$$\boldsymbol{z} = \boldsymbol{d}_{\boldsymbol{f}}(\vec{\boldsymbol{s}}_{\mathrm{t}}) + \boldsymbol{\epsilon}, \tag{4.62}$$

and the cost function (4.54)

$$Q = \boldsymbol{\epsilon}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon} = \left[\boldsymbol{z} - \boldsymbol{d}_{\boldsymbol{f}}(\vec{\boldsymbol{s}})\right]^t \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{z} - \boldsymbol{d}_{\boldsymbol{f}}(\vec{\boldsymbol{s}})\right]. \tag{4.63}$$

The succession (4.55) then becomes

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1} = \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} + (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} [\boldsymbol{z} - \boldsymbol{d}_{\boldsymbol{f}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})], \tag{4.64}$$

where $\boldsymbol{\Sigma} = \mathrm{Cov}\,[\boldsymbol{\epsilon}]$ and the Jacobian $\mathbf{J}_{\mathrm{n}}$ is defined as

$$\mathbf{J}_{\mathrm{n}} \equiv \left.\frac{\partial \boldsymbol{d}_{\boldsymbol{f}}(\vec{\boldsymbol{s}})}{\partial \vec{\boldsymbol{s}}}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} = \begin{pmatrix} \frac{\partial \|\vec{\boldsymbol{r}}_1 - \vec{\boldsymbol{s}}\|_2}{\partial x}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \frac{\partial \|\vec{\boldsymbol{r}}_1 - \vec{\boldsymbol{s}}\|_2}{\partial y}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \\ \cdots & \cdots \\ \frac{\partial \|\vec{\boldsymbol{r}}_{\mathrm{K}} - \vec{\boldsymbol{s}}\|_2}{\partial x}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \frac{\partial \|\vec{\boldsymbol{r}}_{\mathrm{K}} - \vec{\boldsymbol{s}}\|_2}{\partial y}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \end{pmatrix} = \begin{pmatrix} \frac{(\hat{x}_{\mathrm{n}} - r_{\mathrm{x}1})}{d_{1\mathrm{n}}} & \frac{(\hat{y}_{\mathrm{n}} - r_{\mathrm{y}1})}{d_{1\mathrm{n}}} \\ \cdots & \cdots \\ \frac{(\hat{x}_{\mathrm{n}} - r_{\mathrm{x}\mathrm{K}})}{d_{\mathrm{K}\mathrm{n}}} & \frac{(\hat{y}_{\mathrm{n}} - r_{\mathrm{y}\mathrm{K}})}{d_{\mathrm{K}\mathrm{n}}} \end{pmatrix},$$

where $d_{\mathrm{kn}} = \|\vec{\boldsymbol{r}}_{\mathrm{k}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}\|_2$. Defining $\boldsymbol{D}_{\mathrm{n}} = \mathrm{diag}(d_{1\mathrm{n}}, \ldots, d_{\mathrm{K}\mathrm{n}})$ and using (4.41) and (4.42), this expression can be rewritten as

$$\mathbf{J}_{\mathrm{n}} \equiv \boldsymbol{D}_{\mathrm{n}}^{-1} (\mathbf{1} \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}^{\,t} - \boldsymbol{R}). \tag{4.65}$$

with which we can express the covariance of the position estimator (4.60) as

$$\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] = [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} = [(\mathbf{1} \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}^{\,t} - \boldsymbol{R})^t \boldsymbol{D}_{\mathrm{n}}^{-1} \boldsymbol{\Sigma}^{-1} \boldsymbol{D}_{\mathrm{n}}^{-1} (\mathbf{1} \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}^{\,t} - \boldsymbol{R})]^{-1}. \tag{4.66}$$

Comparing this with the CRLB (4.44), we can detect a similar form. In fact, if $\boldsymbol{\Sigma} = \sigma^2 \mathbf{I}$, then both expressions are equal. The difference is that the expression for the CRLB is implicitly evaluated at the node's true position, whereas (4.66) is evaluated at the algorithm's point of convergence.

The linearization error then becomes

$$\boldsymbol{e}_{\mathrm{L}} = [\boldsymbol{d} - \boldsymbol{d}_{\boldsymbol{f}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})] - \mathbf{J}_{\mathrm{n}} (\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}), \tag{4.67}$$

where again $\boldsymbol{d} = \boldsymbol{d}_{\boldsymbol{f}}(\vec{\boldsymbol{s}}_{\mathrm{t}})$. This error depends only on the geometric factors and not on the measurement technology being used.

Additive Gaussian Noise Model

Using the model (4.2) for distance estimation and assuming IID normally distributed errors, we can complete the definition of the model (4.62) in a probabilistic sense as

$$\boldsymbol{z} = \boldsymbol{d_f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) + \boldsymbol{\epsilon}, \text{ where } \boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}). \tag{4.68}$$

The Taylor-series position estimator is given by the succession (4.64), whose bias and covariance of (4.59) and (4.60) simplify respectively to

$$\mathrm{E}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] - \vec{\boldsymbol{s}}_{\mathrm{t}} = (\mathbf{J}_{\mathrm{n}}^t\mathbf{J}_{\mathrm{n}})^{-1}\mathbf{J}_{\mathrm{n}}^t\boldsymbol{e}_{\mathrm{L}} = \boldsymbol{\Phi}_{\mathbf{d}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})\mathbf{J}_{\mathrm{n}}^t\boldsymbol{e}_{\mathrm{L}} \tag{4.69}$$

$$\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] = \sigma^2(\mathbf{J}_{\mathrm{n}}^t\mathbf{J}_{\mathrm{n}})^{-1} = \sigma^2\boldsymbol{\Phi}_{\mathbf{d}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}), \tag{4.70}$$

where we have used (4.65) and the definition of $\boldsymbol{\Phi}_{\mathbf{d}}$ from (4.44). The expression of the covariance of the estimator coincides formally with the CRLB, although in (4.70) $\boldsymbol{\Phi}_{\mathbf{d}}$ it is evaluated at the point of convergence instead of the true position. In any case, the estimator is not necessarily efficient, as it is biased when $\boldsymbol{e}_{\mathrm{L}} \neq \mathbf{0}$.

Log-Normal Model

In Section 4.2.2 we studied the statistical properties of three different distance estimators using RSS measurements and the log-normal model: ML, unbiased and LMMSE. Using any of these estimators as the observation ($\hat{d} = z$) and substituting them in (4.62), we can write the model of the error as

$$\boldsymbol{\epsilon} = \hat{\boldsymbol{d}} - \boldsymbol{d_f}(\vec{\boldsymbol{s}}_{\mathrm{t}}). \tag{4.71}$$

This expression, however, represents a subtle difference with respect to (4.62) and (4.68). In (4.62), the error $\boldsymbol{\epsilon}$ is considered a random variable, which in the case of the additive Gaussian noise model (4.68) is assumed to be normally distributed. As a consequence, the observations, $\boldsymbol{z}$, are a random variable, again normally distributed in the additive Gaussian noise model. When using the log-normal model for distance estimation, we observe directly the outcome of the distance estimators, $\hat{\boldsymbol{d}}$, which are themselves log-normally distributed random variables. As a consequence, the error, defined as the difference between the observation and the true value, is a random variable.

Because the distance estimators in (4.71) are not normally distributed, the errors will not be normally distributed. Moreover, we saw that the ML and LMMSE distance estimators are biased, and therefore $\mathrm{E}[\boldsymbol{\epsilon}] \neq 0$. As a result, the position estimators using these distance estimators will be biased.

We aim now at studying the bias and variance of Taylor-based position estimators using all the previous three distance estimators.

Using the **ML distance estimator** leads to a vector of distance errors $\boldsymbol{\epsilon}_{\text{ML}} = \hat{\boldsymbol{d}}_{\text{ML}} - \boldsymbol{d}$. By virtue of (4.15), we know that $E[\boldsymbol{\epsilon}_{\text{ML}}] = \boldsymbol{d}(\Gamma - 1)$, and from (4.16) and using the assumption of IID samples, we get $\boldsymbol{\Sigma}_{\text{ML}} = \text{Cov}\,[\boldsymbol{\epsilon}_{\text{ML}}] = \Gamma^2(\Gamma^2 - 1)\boldsymbol{D}^2$. The Taylor-series position estimator is then given by the succession (4.64), whose bias and covariance of (4.59) and (4.60) simplify respectively to

$$\text{Bias}\left[\hat{\vec{\boldsymbol{s}}}_{\text{ML}}\right] = \Gamma^2(\Gamma^2 - 1)(\mathbf{J}_{\text{n}}^t\mathbf{J}_{\text{n}})^{-1}\mathbf{J}_{\text{n}}^t\left[\boldsymbol{e}_{\text{L}} + \boldsymbol{d}(\Gamma - 1)\right] \tag{4.72}$$

$$\text{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\text{ML}}\right] = \Gamma^2(\Gamma^2 - 1)[\mathbf{J}_{\text{n}}^t\boldsymbol{D}_{\text{n}}^{-2}\mathbf{J}_{\text{n}}]^{-1} = \Gamma^2(\Gamma^2 - 1)\boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\text{n}}), \tag{4.73}$$

where the definition of $\boldsymbol{\Phi}_{\mathbf{p}}$ is taken from (4.45).

For the **unbiased distance estimator**, the mean of the distance errors vector $\boldsymbol{\epsilon}_{\text{U}} = \hat{\boldsymbol{d}}_{\text{U}} - \boldsymbol{d}$ is zero. From (4.20), and using again the IID assumption of the RSS measurements, the covariance of the distance measurement errors becomes $\boldsymbol{\Sigma}_{\text{U}} = \text{Cov}\,[\boldsymbol{\epsilon}_{\text{U}}] = (\Gamma^2 - 1)\boldsymbol{D}^2$. The bias and covariance of the position estimator are then:

$$\text{Bias}\left[\hat{\vec{\boldsymbol{s}}}_{\text{U}}\right] = (\Gamma^2 - 1)(\mathbf{J}_{\text{n}}^t\mathbf{J}_{\text{n}})^{-1}\mathbf{J}_{\text{n}}^t\boldsymbol{e}_{\text{L}} \tag{4.74}$$

$$\text{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\text{U}}\right] = (\Gamma^2 - 1)[\mathbf{J}_{\text{n}}^t\boldsymbol{D}_{\text{n}}^{-2}\mathbf{J}_{\text{n}}]^{-1} = (\Gamma^2 - 1)\boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\text{n}}). \tag{4.75}$$

The **LMMSE distance estimator** is biased. The error is $\boldsymbol{\epsilon}_{\text{V}} = \hat{\boldsymbol{d}}_{\text{V}} - \boldsymbol{d}$. Using (4.23), we can write $\text{E}\,[\boldsymbol{\epsilon}_{\text{V}}] = (\Gamma^{-2} - 1)\boldsymbol{d}$, and using (4.24) and assuming again IID errors, we arrive at $\boldsymbol{\Sigma}_{\text{V}} = \text{Cov}\,[\boldsymbol{\epsilon}_{\text{V}}] = (\Gamma^{-2} - \Gamma^{-4})\boldsymbol{D}^2$. The bias and covariance of the position estimator are then

$$\text{Bias}\left[\hat{\vec{\boldsymbol{s}}}_{\text{V}}\right] = (\Gamma^2 - 1)(\mathbf{J}_{\text{n}}^t\mathbf{J}_{\text{n}})^{-1}\mathbf{J}_{\text{n}}^t\left[\boldsymbol{e}_{\text{L}} + \boldsymbol{d}(\Gamma^{-2} - 1)\right] \tag{4.76}$$

$$\text{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\text{V}}\right] = (\Gamma^{-2} - \Gamma^{-4})[\mathbf{J}_{\text{n}}^t\boldsymbol{D}_{\text{n}}^{-2}\mathbf{J}_{\text{n}}]^{-1} = (\Gamma^{-2} - \Gamma^{-4})\boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\text{n}}). \tag{4.77}$$

In all cases, the covariance of the position estimators depends on the geometrical arrangement of the beacons and the position of convergence through factor $\boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\text{n}})$, which is then multiplied by a constant that depends on the model parameters. These constants are the same ones that characterize the variance of the respective distance estimators (see (4.16), (4.20) and (4.24)).

4.4.2 Taylor-Series Linearization Estimation for RSS

In the previous subsection we used the sum of squared distance errors as the cost function to be minimized for position estimation. This required a previous distance estimation from RSS measurements. In this subsection we will now use directly the sum of squared differences between the observed and predicted RSS.

Using the log-normal model for RSS predictions, function $\boldsymbol{f}$ in (4.53) becomes

$$\boldsymbol{z_f}(\vec{\boldsymbol{s}}) \equiv \boldsymbol{f}(\vec{\boldsymbol{s}}) = \begin{pmatrix} \bar{z}_1 \\ \vdots \\ \bar{z}_{\mathrm{K}} \end{pmatrix}, \tag{4.78}$$

with $\bar{z}_{\mathrm{k}}$ being defined in (4.5). The Jacobian (4.56) then becomes

$$\mathbf{J}_{\boldsymbol{z}n} \equiv \left.\frac{\partial \boldsymbol{z_f}(\vec{\boldsymbol{s}})}{\partial \vec{\boldsymbol{s}}}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} = \begin{pmatrix} \frac{\partial \bar{z}_1}{\partial x}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \frac{\partial \bar{z}_1}{\partial y}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \\ \vdots & \vdots \\ \frac{\partial \bar{z}_K}{\partial x}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \frac{\partial \bar{z}_K}{\partial y}\big|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \end{pmatrix} = -\frac{10\,n}{\ln 10} \begin{pmatrix} \frac{(\hat{x}_{\mathrm{n}} - r_{\mathrm{x}1})}{d_{1\mathrm{n}}^2} & \frac{(\hat{y}_{\mathrm{n}} - r_{\mathrm{y}1})}{d_{1\mathrm{n}}^2} \\ \vdots & \vdots \\ \frac{(\hat{x}_{\mathrm{n}} - r_{\mathrm{x}\mathrm{K}})}{d_{\mathrm{K}\mathrm{n}}^2} & \frac{(\hat{y}_{\mathrm{n}} - r_{\mathrm{y}\mathrm{K}})}{d_{\mathrm{K}\mathrm{n}}^2} \end{pmatrix}$$

which, using (4.65), can be rewritten as

$$\mathbf{J}_{\boldsymbol{z}n} = -\frac{10\,n}{\ln 10}\boldsymbol{D}_{\mathrm{n}}^{-2}(\mathbf{1}\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}^{\,t} - \boldsymbol{R}) = -\frac{10\,n}{\ln 10}\boldsymbol{D}_{\mathrm{n}}^{-1}\mathbf{J}_{\mathrm{n}}.$$

The measurement errors are IID with $e_k \sim N(0, \sigma^2)$, and thus the covariance matrix of the errors is $\mathrm{Cov}\,[\boldsymbol{\epsilon}] = \sigma^2 \boldsymbol{I}$. The Taylor-series position estimator is given by the succession (4.64), whose bias and covariance of (4.59) and (4.60) simplify respectively to

$$\mathrm{E}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] - \vec{\boldsymbol{s}}_{\mathrm{t}} = (\mathbf{J}_{\boldsymbol{z}n}^t \mathbf{J}_{\boldsymbol{z}n})^{-1}\mathbf{J}_{\boldsymbol{z}n}^t \boldsymbol{e}_{\mathbf{Lz}} \tag{4.79}$$

$$\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] = \sigma^2[\mathbf{J}_{\boldsymbol{z}n}^t \mathbf{J}_{\boldsymbol{z}n}]^{-1} = \gamma^2[(\mathbf{1}\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}^{\,t} - \boldsymbol{R})^t \boldsymbol{D}_{\mathrm{n}}^{-4}(\mathbf{1}\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}^{\,t} - \boldsymbol{R})]^{-1}, \tag{4.80}$$

where the linearization error now is

$$\boldsymbol{e}_{\mathbf{Lz}} = \left[\boldsymbol{z_f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) - \boldsymbol{z_f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})\right] - \mathbf{J}_{\boldsymbol{z}n}(\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}). \tag{4.81}$$

Using (4.65) and the definition of $\boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})$ from (4.45), then (4.79) and (4.80) can be rewritten as

$$\begin{aligned} \mathrm{E}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] - \vec{\boldsymbol{s}}_{\mathrm{t}} &= -\left(\frac{\ln 10}{10n}\right)^3 [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{D}_{\mathrm{n}}^{-2}\mathbf{J}_{\mathrm{n}}]^{-1}\mathbf{J}_{\mathrm{n}}^t \boldsymbol{D}_{\mathrm{n}}^{-1}\boldsymbol{e}_{\mathbf{Lz}} \\ &= -\left(\frac{\ln 10}{10n}\right)^3 \boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})\mathbf{J}_{\mathrm{n}}^t \boldsymbol{D}_{\mathrm{n}}^{-1}\boldsymbol{e}_{\mathbf{Lz}} \\ \mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] &= \gamma^2[\mathbf{J}_{\mathrm{n}}^t \boldsymbol{D}_{\mathrm{n}}^{-2}\mathbf{J}_{\mathrm{n}}]^{-1} = \gamma^2 \boldsymbol{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}). \end{aligned} \tag{4.82}$$

As we see, the expression of the covariance has the same form as the CRLB of (4.45), although now it is evaluated at the position of convergence.

4.4.3 Comparison Among RSS-Based Methods

We have studied thus far four different RSS-based position estimation methods using NLS optimization, with all of them having different statistical properties. In the three first methods the cost function penalizes the differences between the observed (estimated) and true node-beacon distances, for which it is necessary to estimate the distances from RSS measurements in a preliminary step. In the fourth method, the penalization is straight with respect to the difference between the observed and predicted RSS measurements, without using distance estimation in an intermediate step. We will now compare the different methods.

The bias of the different localization methods is the result of two factors: the bias of the errors in the measurement model and the linearization error. Even in cases with unbiased measurement errors, the presence of the linearization error makes the position estimators biased, and therefore inefficient. The amount of bias introduced by the linearization error depends on how strong the non-linearity is around the convergence point and how far it is from the true position. The convergence position depends on the shape of the cost function, which is different among the position estimators being considered, even when the observations are the same. Thus, an analytical comparison among the bias of the different localization methods due to the linearization error is not straightforward.

The covariance of each of the position estimators takes the form $\kappa\,\mathbf{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})$ (see (4.73),(4.75),(4.77) and (4.82)), where recall that $\mathbf{\Phi}_{\mathbf{p}}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})$ depends exclusively on the problem geometry and κ is a constant that depends only on the model parameters through γ and assumes different values for the different localization methods being used. Figure 4.3 shows this factor as a function of γ for the different position estimators and for $\gamma \in [0, 1.5]$. The factor κ_{P} corresponds to the method using RSS measurements directly, whose formula for the covariance coincides with the CRLB. Clearly, for the same position estimate the algorithms

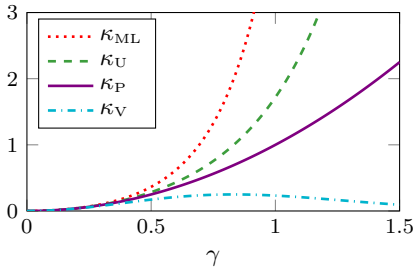


Figure 4.3: Comparison of κ-factors among the different Taylor-series linearization-based localization algorithms.

using ML and unbiased distance estimators always have greater covariance for any value of γ (κ_{ML} and κ_{U} respectively), whereas for the algorithm using the LMMSE distance estimator the covariance is lower (κ_{V}).

In practice, due to the difficulty to predict the convergence position of the different algorithms and therefore the bias, we cannot assess analytically which one performs better in terms of the MSE, even if the comparison of the covariance seems straightforward. Recognizing this limitation we aim at a comparison based on computer simulations.

Simulation Setup

Because the performance of the algorithms depends on the geometrical configuration of the problem, assessing the performance in all possible cases requires placing a target node in all possible geometrical configurations, generating a large amount of random measurements for each case and computing the statistics for the resulting position estimates for each of the configurations. This requires solving an enormous amount of localization problems and, hence, is not practical. Instead, we are interested in a practical comparison from which we can derive some general guidelines. The problem now is how to come up a generic enough simulation setup that represents a wide range of possible use cases so that the simulations become computationally tractable.

In particular we are interested in comparing the algorithms in situations with good and bad geometrical conditions, that is, with good and bad GDoP. Intuitively, good geometrical conditions occur when the target node falls within the polygonal area defined by the beacons, whereas bad conditions occur when the node is outside. In our simulation scenario we place ten beacons equidistant from the origin of coordinates at a distance of 15 m and all equiangularly spaced. We then place target nodes in positions within a disk of radius 30 m so that the nodes fall both inside and outside of the polygonal area defined by the beacons.

To place the target nodes, first the disk is divided into 60 concentric and non-overlapping annuli centred also in the origin of coordinates, with the internal and external radii of each annuli being $k\Delta d$ and $(k+1)\Delta d$, respectively, and with $\Delta d = 0.5$ m and $k = \{0, \ldots, 59\}$. In each of the annuli we place 500 target nodes in uniformly distributed random positions. Therefore, we have a total of $3 \cdot 10^4$ nodes in the whole disk. Figure 4.4 presents a simplified schema of this arrangement. The small circles symbolize the nodes placed in their respective annuli. The nodes in dark blue correspond to two sample annuli, one inside and the other outside the polygonal area defined by the beacons.

For each of the nodes, we generate one unique vector of noisy RSS measurements using the log-normal model. This vector is then used to locate the corresponding target node using the different localization algorithms to be compared. In all cases the true position was used as the initial value to start the

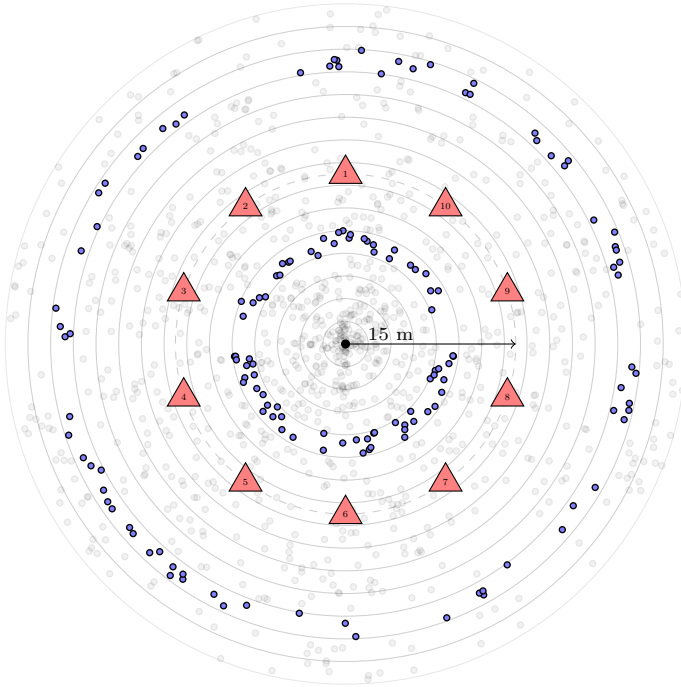


Figure 4.4: Simplified schema of simulation setup.

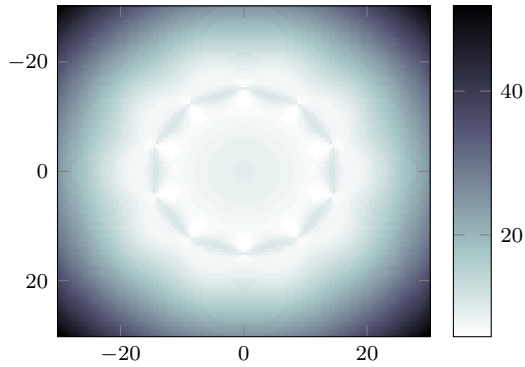


Figure 4.5: Characteristic error (γ-normalized) of the simulation setting.

solver. Once the position estimates are ready we calculate the average distance error for the nodes belonging to the same annuli. [4] Then, we plot the error statistics for the different algorithms as a function of the radii of the central circle within the annulus.

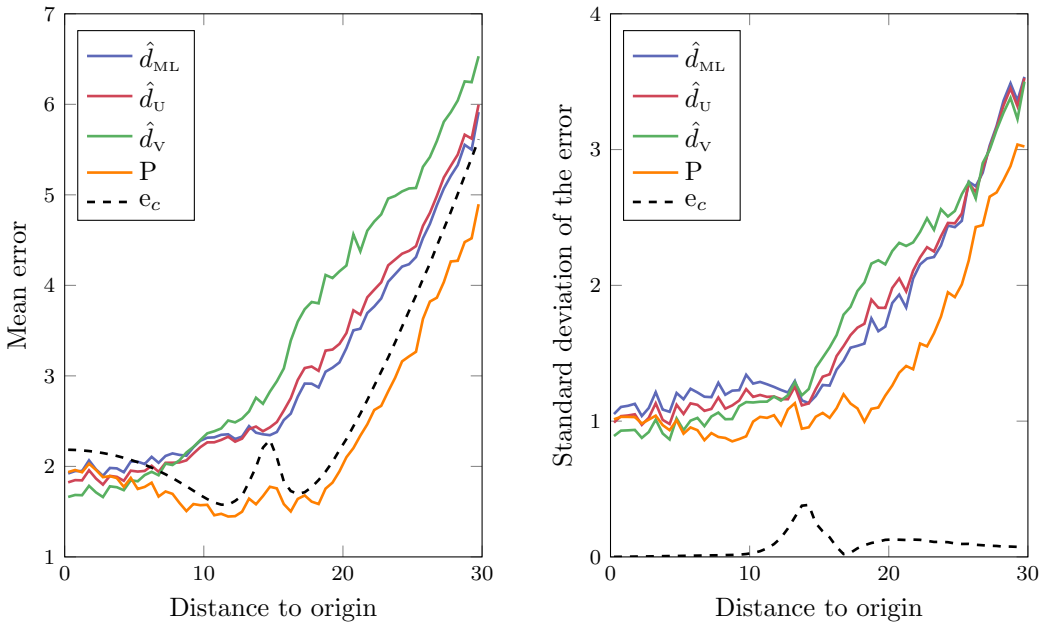
Placing many beacons at equidistant positions from the origin of coordinates and with equiangular separation creates a special scenario in which nodes within the same annuli will see a rather similar geometry of the problem. This can be seen more clearly in Figure 4.5, which represents the characteristic error for any node position within the proposed simulation scenario normalized with γ. We define the characteristic error as $e_c = \sqrt{\mathrm{Tr}(\mathrm{CRLB})}$, that is, the square root of the trace of the CRLB. This is a measure of the typical error of the best unbiased estimators in metres, and thus it can be used to assess the quality of the estimation methods. Normalizing it with γ makes it independent of the model parameters and dependent only on the geometrical conditions of the problem (see (4.45)). Thus, e_c/γ is a measure of GDoP. Errors corresponding to points with similar intensity in Figure 4.5 will be affected similarly by the geometrical configuration of the problem.

Although the geometrical condition of the problem for nodes within the same annulus is not exactly the same, they are relatively similar, except perhaps closer to the 15 m radius where the beacons are located. This allows us to compute the error corresponding to each annulus as the average of the errors from its belonging nodes. This comparison is not strictly valid for all possible configurations, but we believe that it gives us a reasonable framework within which we can make general comparisons of the different algorithms.

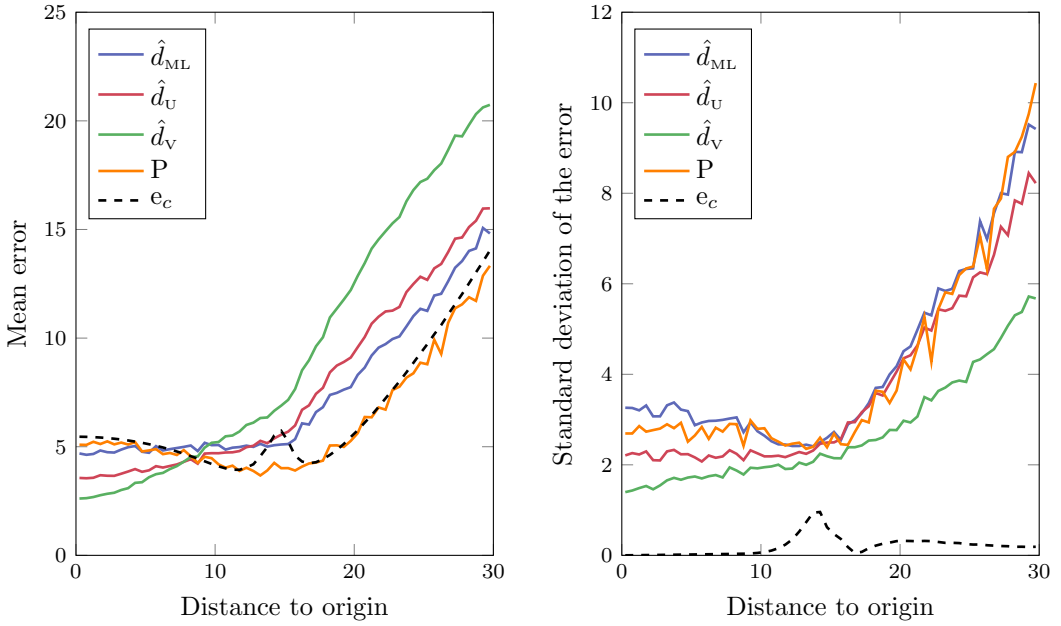
Simulation Results

Figure 4.6 shows the average and standard deviation of the errors for the different localization methods. The horizontal axes present the radius of the central circle of the annuli, which corresponds approximately to the distance from the nodes to the origin of coordinates. Results associated with using distance estimates are indicated with a $\hat{d}$ in the legend and a subindex referring to the distance estimator being used. Results associated with using directly the RSS measurements are indicated with the letter P (from power). Panel 4.6a uses $\sigma = 2$, and Panel 4.6b $\sigma = 5$. The figures also show the statistics of the characteristic error. Its standard deviation is only a guide to assess the variability of the GDoP within each annuli.

[4] This average is calculated only using the position estimates that converged to a solution.

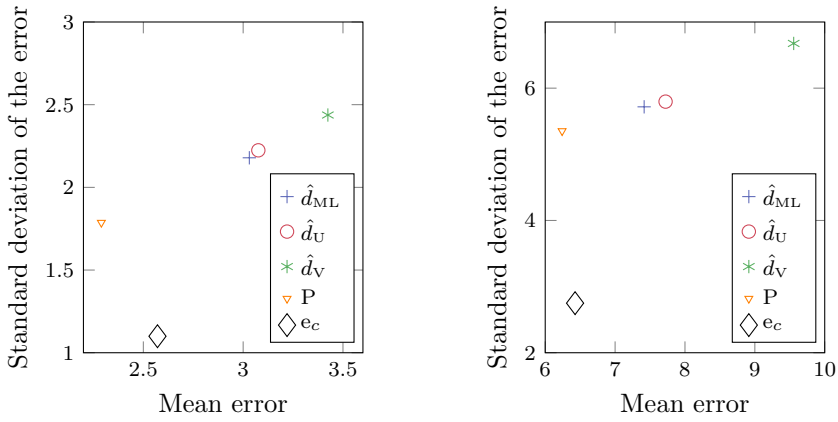


(a) Average and standard deviation of the error for $\sigma = 2$



(b) Average and standard deviation of the error for $\sigma = 5$

Figure 4.6: Comparison of Taylor methods. Model parameters: $z_0 = -70$, $n = 2$.



(a) $\sigma = 2$ (b) $\sigma = 5$

Figure 4.7: Global comparison of the Taylor methods. Model parameters: $z_0 = -70$, $n = 2$, $\sigma = 2$ (left) and $\sigma = 5$ (right).

As we can see in Panel 4.6a, for relatively small noise power the method that directly uses RSS performs significantly better in general. It follows the trend of the normalized characteristic error, with a slightly decaying average error until a distance of 15 m (where the beacons are located) and a subsequent increase for larger distances. In contrast, the mean error associated with the distance-based methods increases from the zero distance. Among these three estimators, the LMMSE seems to be slightly better within the area of good GDoP, but this behaviour changes in the area of bad GDoP. On the right-hand side of the figure, we can also see that the standard deviation of the error is comparable among the distance-based methods, and it is lowest in general for the direct RSS algorithm.

The difference in performance increases with the noise power (see Panel 4.6b). Within the area of good GDoP, the method using LMMSE distance estimators is generally superior, with a lower mean and standard deviation of the error. However this error increases rapidly with the distance to the origin, and becomes highest in the area of bad GDoP. Although the variability in the error is smallest in all the cases (see the right side of the figure), the results suggest that the bias becomes large enough to make the mean error comparatively high with respect to the other methods. The method that directly uses RSS still behaves similarly to the characteristic error, and gives the best estimates in situations with bad GDoP.

Figure 4.7 presents an aggregated comparison of all the methods in our simulation scenario. The horizontal axes show the mean error of all the position

estimates for the respective algorithms, while the vertical axes the standard deviation. The better the estimator performs globally, the closer it will be to the lower left corner of the figure. As we can see, the direct RSS-based algorithm performs better in general in cases with both low and higher levels of noise power. In a very general case in which we do not know whether the geometry of the problem is good or bad, it seems reasonable to use the direct RSS method. The unbiased and LMMSE methods might be better for cases in which we can ensure a good GDoP, especially with higher noise power.

4.5 Linear LS Methods

The NLS estimation methods from the previous section rely on iterative solvers that require a Taylor-series linearization of the measurement model. Another solution is to find a mathematical transformation of the equations so that the resulting formulas are linear in the coordinates. Once the equations are linear, we can use different techniques to estimate the nodes' position.

In some particular cases one can find a transformation of equations (4.1) that lead to an analytical solution of the problem even with noisy distance measurements. The particular case of having three distance measurements available (trilateration) has received special attention in the literature, especially in the field of aviation since three is the minimum number of beacons needed to estimate the 3D position of an aircraft.[5] For this particular configuration, researchers have tried to come up with an analytical solution for the position estimate [Fang, 1986, Manolakis, 1996, Thomas and Ros, 2005].

The solution proposed by Manolakis in [Manolakis, 1996] is based on the following simple transformation: first select one of the beacons whose corresponding equation will then be subtracted from the others. In the following, for clarity we denote this beacon as pivot. Similarly, we also denote as pivot its corresponding equation and measurement. From (4.1), using the last equation as pivot and rearranging terms after the subtraction, we can obtain the following linear system of equations:

$$\begin{bmatrix} (r_{\mathrm{x3}} - r_{\mathrm{x1}}) & (r_{\mathrm{yK}} - r_{\mathrm{y1}}) \\ (r_{\mathrm{x3}} - r_{\mathrm{x2}}) & (r_{\mathrm{yK}} - r_{\mathrm{y2}}) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \begin{bmatrix} (d_1^2 - d_3^2) + (r_3^2 - r_1^2) \\ (d_2^2 - d_3^2) + (r_3^2 - r_2^2) \end{bmatrix} \tag{4.83}$$

where recall that $\vec{\boldsymbol{r}}_{\mathrm{k}} = (r_{\mathrm{xk}}, r_{\mathrm{yk}})^t$ and we define $r_{\mathrm{k}} \equiv \|\vec{\boldsymbol{r}}_{\mathrm{k}}\|$. Although here we present the transformation for a two-dimensional coordinate system, Manolakis applied this transformation to a problem that includes the altitude. He provided exact analytical formulas for the horizontal and vertical position and studied the

[5] Note that the ambiguity in the altitude is automatically resolved by discarding the position with a negative altitude.

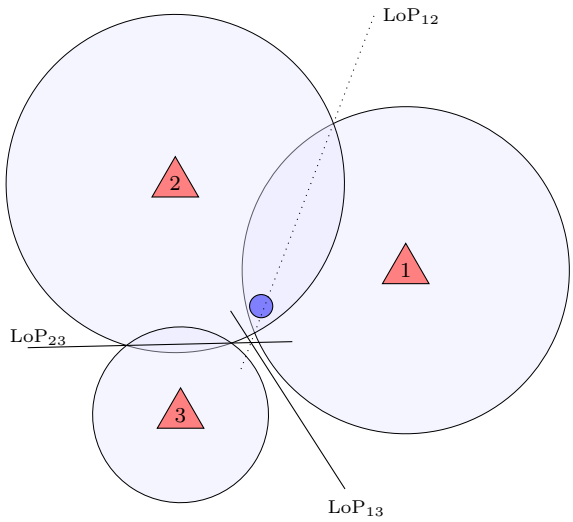


Figure 4.8: Lines of position derived from the exact linearization transformation.

performance of the proposed estimator providing approximated formulas for its bias and covariance. Additionally, he calculated the GDoP and vertical dilution of precision (VDoP) for an equidistant triangular arrangement of the beacons.

Later, Caffery provided an intuitive geometrical interpretation of the transformation, which converts three equations representing circles and their intersections into two representing lines and their intersections [Caffery, 2000]. These lines are referred to as LoP. Thus, the transformed system of equations (4.83) represents the LoPs associated with circles 1-3 and 2-3 respectively. When the corresponding two circles intersect, the associated LoP will pass through the two points of intersection. If they are tangent, the corresponding LoP will pass through the tangent point. And even if the circles do not intersect, the transformation leads to a proper line that can be used for localization.

One key aspect of this transformation is then that, while the presence of noisy distance measurements renders the original system of equations inconsistent, after the transformation the system can naturally accommodate the presence of noise. We can see this in Figure 4.8, which presents the same situation as Figure 4.1 but with noisy measurements. LoP_{13} and LoP_{23} are associated with circles 1-3 and 2-3, respectively. As we can see, LoP_{23} passes through the points of intersection of its associated circles. Circles 1 and 3 do not intersect, and thus (4.1) would yield an inconsistent system of equations. Instead, the transformation yields LoP_{13}, which intersects with LoP_{23}.

In the derivation of (4.83), we used the third beacon as pivot, and therefore subtracted the equation for its associated circle from the other two. The point of intersection does not change if we choose either of the other two beacons as pivots. Figure 4.8 shows the three possible LoPs and how they all intersect in the same point: the estimated position. Thus, for the particular case in which there are three beacons, we have an estimator with an analytical and unique solution. Should we have more than three beacons, we can still apply the previous transformation, but no general exact analytical solution is known.

In the more general case in which we have $K \geq 3$ beacons, subtracting the i-th equation from the others in (4.1) and rearranging terms yields the following linear system of equations:

$$\underbrace{\begin{bmatrix} (r_{x_i} - r_{x_1}) & (r_{y_i} - r_{y_1}) \\ \cdots & \cdots \\ (r_{x_i} - r_{x_{i-1}}) & (r_{y_i} - r_{y_{i-1}}) \\ (r_{x_i} - r_{xi+1}) & (r_{y_i} - r_{y_{i+1}}) \\ \cdots & \cdots \\ (r_{x_i} - r_{x_K}) & (r_{y_i} - r_{y_K}) \end{bmatrix}}_{\boldsymbol{A_L}} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2} \underbrace{\begin{bmatrix} (d_1^2 - d_i^2) + (r_i^2 - r_1^2) \\ \cdots \\ (d_{i-1}^2 - d_i^2) + (r_i^2 - r_{i-1}^2) \\ (d_{i+1}^2 - d_i^2) + (r_i^2 - r_{i+1}^2) \\ \cdots \\ (d_K^2 - d_i^2) + (r_i^2 - r_K^2) \end{bmatrix}}_{\boldsymbol{b_L}}. \tag{4.84}$$

Caffery studied possible solutions for problems with more than three beacons. Based on the uniqueness of the solution when only three beacons are available, he proposed two types of methods to solve problems with $K > 3$. The first solution tries to find the intersections of the possible LoPs that can be formed with the set of K circles. The position estimate is then the mean of the intersection points or the centroid of the polygon that they form. The second solution applies LS directly to (4.84).

Caffery worked only with ToA localization, and therefore he used the AGN model of (4.2). Using simulations, he concluded that the LS-based solution was '*slightly worse*' than using Taylor linearization estimation, but was '*significantly better*' in the presence of large bias. The exact linearization showed almost the same results as the Taylor series linearization method presented in the previous section when three beacons were used. The results were different when using 4 beacons, being the Taylor expansion better. This difference is highly relevant when the target devices to be located are outside the central region between the beacons. This emphasizes the effect of the geometry of the problem [Caffery, 2000].

Chitte used the log-normal model for RSS-based distance estimation and applied OLS to solve (4.84), calculating the bias and variance of the resulting estimators [Chitte et al., 2009a, Chitte et al., 2009b]. In the following, we build on and complete the work by Caffery and Chitte by providing more insights and two new families of linear position estimators, namely GLS and IGLS.

4.5.1 OLS

The matrix $\boldsymbol{A}_{\text{L}}$ in (4.84) depends exclusively on the position of the beacons. The vector $\boldsymbol{b}_{\text{L}}$ is the measurement vector of the linearized system, which is a transformation of the original vector of distances. Because the true distances are unknown, in practice we substitute them by some estimates, either of distances or square distances. This leads to using the estimate $\hat{\boldsymbol{b}}_{\text{L}}$ instead of the true $\boldsymbol{b}_{\text{L}}$, which in general will be a random vector that depends on the original distances. For a generic $\hat{\boldsymbol{b}}_{\text{L}}$, the OLS position estimator is given by

$$\hat{\vec{\boldsymbol{s}}} = \underbrace{[\boldsymbol{A}_{\text{L}}^t \boldsymbol{A}_{\text{L}}]^{-1} \boldsymbol{A}_{\text{L}}^t}_{\boldsymbol{C}_{\text{L}}} \hat{\boldsymbol{b}}_{\text{L}} = \boldsymbol{C}_{\text{L}} \hat{\boldsymbol{b}}_{\text{L}}. \tag{4.85}$$

Its expectation and covariance are

$$\begin{aligned} \text{E}\left[\hat{\vec{\boldsymbol{s}}}\right] &= \boldsymbol{C}_{\text{L}}\ \text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] \\ \text{Cov}\left[\hat{\vec{\boldsymbol{s}}}\right] &= \boldsymbol{C}_{\text{L}}\ \text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right]\ \boldsymbol{C}_{\text{L}}^t = \boldsymbol{C}_{\text{L}}\ \boldsymbol{\Sigma}_{\text{L}}\ \boldsymbol{C}_{\text{L}}^t, \end{aligned} \tag{4.86}$$

where we define

$$\boldsymbol{\Sigma}_{\text{L}} \equiv \text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right]. \tag{4.87}$$

Being a transformation of a vector of estimates, the statistical properties of $\hat{\boldsymbol{b}}_{\text{L}}$ will depend on those of the original vector of estimates, as well as on the functional form of the dependence.

AGN model

When using the **additive Gaussian error model** (4.2), we can substitute the distance estimates (4.3) for the true distances in (4.84), which leads to

$$\hat{\boldsymbol{b}}_{\text{L}} = \frac{1}{2} \begin{bmatrix} (\hat{d}_1^2 - \hat{d}_{\text{i}}^2) + (r_{\text{i}}^2 - r_1^2) \\ \cdots \\ (\hat{d}_{\text{i}-1}^2 - \hat{d}_{\text{i}}^2) + (r_{\text{i}}^2 - r_{\text{i}-1}^2) \\ (\hat{d}_{\text{i}+1}^2 - \hat{d}_{\text{i}}^2) + (r_{\text{i}}^2 - r_{\text{i}+1}^2) \\ \cdots \\ (\hat{d}_{\text{K}}^2 - \hat{d}_{\text{i}}^2) + (r_{\text{i}}^2 - r_{\text{K}}^2) \end{bmatrix}. \tag{4.88}$$

The squaring operation on the distance estimators also affects the errors of the model, and thus $\hat{\boldsymbol{b}}_{\text{L}}$ will have a different distribution with respect to the vector of

distance estimates. It can be shown that its expectation and covariance are (see Appendix A.3 for details):

$$\begin{aligned} \mathrm{E}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] &= \boldsymbol{b}_{\mathbf{L}} \\ \mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] &= \sigma^2(\boldsymbol{D}_{\mathbf{L}}^2 + d_{\mathrm{i}}^2 \mathbf{1}_{\mathbf{L}}\mathbf{1}_{\mathbf{L}}^t) + \frac{\sigma^2}{2}(\mathbf{I} + \mathbf{1}_{\mathbf{L}}\mathbf{1}_{\mathbf{L}}^t), \end{aligned} \tag{4.89}$$

where $\boldsymbol{D}_{\mathbf{L}}$ and $\mathbf{1}_{\mathbf{L}}$ have reduced dimensions, that is to say,

$$\boldsymbol{D}_{\mathbf{L}} = \begin{pmatrix} d_1 & & & & \\ & \ddots & & \mathbf{0} & \\ & & d_{\mathrm{i}-1} & & \\ & & & d_{\mathrm{i}+1} & \\ & \mathbf{0} & & & \ddots \\ & & & & & d_{\mathrm{K}} \end{pmatrix} \in \mathbb{R}^{(K-1)\times(K-1)} \tag{4.90}$$

$$\mathbf{1}_{\mathbf{L}} = \begin{pmatrix} 1 & \dots & 1 \end{pmatrix}^t \in \mathbb{R}^{(K-1)\times 1}. \tag{4.91}$$

Because $\hat{\boldsymbol{b}}_{\mathrm{L}}$ is unbiased, the OLS position estimator will also be unbiased.

Log-normal model

When using RSS measurements under the log-normal model, instead of using the squares of the distance estimators we can directly use the estimators of the squared distances that we studied in Subsection 4.2.3. The vector $\hat{\boldsymbol{b}}_{\mathrm{L}}$ then takes a slightly different form with respect to (4.88), being

$$\hat{\boldsymbol{b}}_{\mathrm{L}} = \frac{1}{2} \begin{bmatrix} (\widehat{d_1^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_1^2) \\ \dots \\ (\widehat{d_{\mathrm{i}-1}^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_{\mathrm{i}-1}^2) \\ (\widehat{d_{\mathrm{i}+1}^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_{\mathrm{i}+1}^2) \\ \dots \\ (\widehat{d_{\mathrm{K}}^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_{\mathrm{K}}^2) \end{bmatrix}. \tag{4.92}$$

When using the **ML distance estimator** of d^2, (4.26), the expectation and

covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ become (see Appendix A.4)

$$\text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{2}\begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix} + \boldsymbol{b}_{\text{L}} \tag{4.93}$$

$$\text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{4}\Gamma_2^2(\Gamma_2^2 - 1)(\boldsymbol{D}_{\text{L}}^4 + d_{\text{i}}^4 \mathbf{1}_{\text{L}}\mathbf{1}_{\text{L}}^t). \tag{4.94}$$

The OLS position estimator is thus biased, with expectation

$$\text{E}\left[\hat{\vec{\boldsymbol{s}}}\right] = \vec{\boldsymbol{s}} + \frac{1}{2}(\Gamma_2 - 1)\boldsymbol{C}_{\text{L}}\begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix}. \tag{4.95}$$

When using the **unbiased squared distance estimator** (4.32) (m=2), the expectation and covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ become (see Appendix A.4)

$$\begin{aligned} \text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] &= \boldsymbol{b}_{\text{L}} \\ \text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] &= \frac{1}{4}(\Gamma_2^2 - 1)\big(\boldsymbol{D}_{\text{L}}^4 + d_{\text{i}}^4\, \mathbf{1}_{\text{L}}\mathbf{1}_{\text{L}}^t\big). \end{aligned} \tag{4.96}$$

The position estimator is then unbiased.

Finally, when using the **LMMSE squared distance estimator** (4.35) (m=2), the expectation and covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ become (see Appendix A.4)

$$\text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{2}(\Gamma_2^{-2} - 1)\begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix} + \boldsymbol{b}_{\text{L}} \tag{4.97}$$

$$\text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{4}\frac{\Gamma_2^2 - 1}{\Gamma_2^4}\big(\boldsymbol{D}_{\text{L}}^4 + d_{\text{i}}^4\, \mathbf{1}_{\text{L}}\mathbf{1}_{\text{L}}^t\big), \tag{4.98}$$

which also leads to a biased OLS position estimator, with expectation

$$\mathrm{E}\left[\hat{\vec{s}}\right] = \vec{s} + \frac{1}{2}(\Gamma_2^{-2} - 1)\boldsymbol{C}_{\mathbf{L}} \begin{pmatrix} d_1^2 - d_i^2 \\ \cdots \\ d_{i-1}^2 - d_i^2 \\ d_{i+1}^2 - d_i^2 \\ \cdots \\ d_K^2 - d_i^2 \end{pmatrix}. \tag{4.99}$$

Comments

In contrast to the NLS estimation methods studied in the previous section, the covariance $\boldsymbol{\Sigma}_{\mathbf{L}} = \mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right]$ has, in general, dissimilar elements in the diagonal, which is synonymous with heteroscedasticity. Furthermore, the non-diagonal terms are non-zero, which indicates presence of correlation among the transformed measurements despite the IID errors assumption. This correlation is a direct consequence of subtracting the pivot equation from the rest when applying the transformation to the original distance measurements. As a result, all the non-diagonal terms are equal.

When using LLS position estimators, the geometry of the problem directly affects the covariance of the transformed measurements vector, $\hat{\boldsymbol{b}}_{\mathrm{L}}$, which in turn affects the covariance of the position estimators. From (4.89),(4.94),(4.96) and (4.98), we can see that $\boldsymbol{\Sigma}_{\mathbf{L}}$ increases with any and all of the node-beacon distances. This increase is with the fourth power when using RSS measurements. This suggests that, when deploying beacons, one should distribute them so that the node-beacon distances are minimized. Such is the case with a symmetric distribution trying to maximize the probability of the target nodes being in the centre. Regarding the pivotal beacon, its contribution is additive for all the elements of $\boldsymbol{\Sigma}_{\mathbf{L}}$. This suggests that choosing the beacon closest to the target node as pivot will reduce the covariance of its position estimate.

The geometry of the problem also directly affects the bias when the position estimators are biased, that is to say, when using ML and LMMSE distance estimators under the log-normal model (see (4.95) and (4.99)). The selection of the pivotal beacon will affect both $\boldsymbol{C}_{\mathbf{L}}$ and the vector of differences of squared distances, and therefore it is clear that it can affect the bias. However, unlike the case of $\boldsymbol{\Sigma}_{\mathbf{L}}$, the impact of this selection is not immediately clear.

Our choice is to select as pivot the beacon closest to the target node, because a) as we have seen this minimizes the covariance of the position estimator, and b) as we will see later, it results in a more numerically stable implementation of the algorithm. Because the true position of the node is not known, the pivotal beacon is selected as the one whose estimated distance to the node is the smallest.

4.5.2 GLS and IGLS

A consequence of $\boldsymbol{\Sigma}_{\mathbf{L}} \neq \sigma^2 \boldsymbol{I}$ is that the OLS solution is not efficient. In such cases, the GLS is the efficient linear estimator [Greene, 2011, p.264]. The problem is that $\boldsymbol{\Sigma}_{\mathbf{L}}$ is unknown because, as we have just seen in the previous subsection, it depends on the true node-beacon distances, and therefore, on the position that we are trying to estimate.

Nevertheless, we can compute an estimate $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ using the distances or square distance estimates that we used for the OLS solution. In order to do so, when using the AGN model we substitute the estimated distances, $\hat{d}_{\mathrm{k}}$, for the true ones ,d_{k}, in (4.90) and (4.89). Under the log-normal model, we again use instead direct estimates of the squared distances, $\widehat{d^2_{\mathrm{k}}}$. Substituting them for the true distances, d_{k}, in (4.90) leads to calculating $\boldsymbol{D}^2_{\mathbf{L}}$ instead of $\boldsymbol{D}_{\mathbf{L}}$. This matrix will be used in the respective equations (4.94), (4.96) and (4.98) to calculate $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$, in which it has to be squared instead of exponentiated to the fourth power.

Using $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ instead of $\boldsymbol{\Sigma}_{\mathbf{L}}$ leads to the following GLS type of estimator:

$$\hat{\vec{\boldsymbol{s}}} = [\boldsymbol{A}^t_{\mathbf{L}} \hat{\boldsymbol{\Sigma}}^{-1}_{\mathbf{L}} \boldsymbol{A}_{\mathbf{L}}]^{-1} \boldsymbol{A}^t_{\mathbf{L}} \hat{\boldsymbol{\Sigma}}^{-1}_{\mathbf{L}} \hat{\boldsymbol{b}}_{\mathrm{L}}, \tag{4.100}$$

where

$$\hat{\boldsymbol{\Sigma}}_{\mathbf{L}} = f(\widehat{\boldsymbol{d}}_m, \boldsymbol{\theta})$$
$$\widehat{\boldsymbol{d}}_m \equiv (\widehat{d^m_1}, \ldots, \widehat{d^m_{\mathrm{K}}})^t.$$

Furthermore, once we have calculated the position as in (4.100), the node-beacon distances can be recalculated using the new position estimate as the true node position, with which we can reestimate $\widehat{\boldsymbol{d}}_m$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$. This iteration can be repeated until convergence. We refer to this solution as IGLS.

The IGLS algorithm is presented in Table 4.2. It accepts as inputs the matrix $\boldsymbol{R} = [\vec{\boldsymbol{r}}_1, \vec{\boldsymbol{r}}_2, \ldots, \vec{\boldsymbol{r}}_{\mathrm{K}}]^t$ containing the position of the beacons, the vector of initial distance power estimates, $\widehat{\boldsymbol{d}}_m$, and the model parameters, $\boldsymbol{\theta}$. The calculation of $\widehat{\boldsymbol{d}}_m$ depends on whether we consider the AGN or the log-normal model. Under the AGN model we use $(\widehat{d}_{\mathrm{k}})^m$ as the estimator of d^m_{k}, whereas under the log-normal model we use estimates of the distance powers. In line 2 the matrix $\boldsymbol{A}_{\mathbf{L}}$ is calculated as in (4.84), which depends only on the positions of the beacons and is constant. In line 3 the transformed measurement vector, $\hat{\boldsymbol{b}}_{\mathrm{L}}$, is calculated using (4.92), which depends both on $\widehat{\boldsymbol{d}}_m$ and $\boldsymbol{R}$. In line 5, $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ is calculated using the corresponding formula for $\mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right]$ as explained before, which depends on $\widehat{\boldsymbol{d}}_m$ and $\boldsymbol{\theta}$ (more precisely on n and σ from $\boldsymbol{\theta}$ under the log-normal model). Then, in line 6 the GLS position estimate is computed using $\boldsymbol{A}_{\mathbf{L}}$, $\hat{\boldsymbol{b}}_{\mathrm{L}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ as calculated previously. Using this position estimate we can compute a new estimate of the node-beacon distances, with which we re-calculate $\widehat{\boldsymbol{d}}_m$ simply by exponentiating

Table 4.2: Iterative GLS algorithm (IGLS).

```
 1  iglse(R, d̂_m, θ)
 2    A_L = f(R)
 3    b̂_L = f(d̂_m, R)
 4    repeat
 5      Σ̂_L = f(d̂_m, θ)
 6      ŝ⃗ = [A_L^t Σ̂_L^{-1} A_L]^{-1} A_L^t Σ̂_L^{-1} b̂_L
 7      d̂_m = (||ŝ⃗ − r⃗_1||^m, ..., ||ŝ⃗ − r⃗_K||^m)
 8    until convergence
 9    Σ̂_L = f(d̂_m, θ)
10    Cov[ŝ⃗] = [A_L^t Σ̂_L^{-1} A_L]^{-1}
11    return (ŝ⃗, Cov[ŝ⃗])
```

them. The new value of $\widehat{\boldsymbol{d}}_m$ will be used in the next iteration to re-calculate $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$. Steps 5 to 7 are repeated until a convergence criterion is satisfied. Finally, the covariance of the position estimate is approximated in line 10 (see below), for which $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ is re-calculated in line 9.

The study of the statistical properties of the GLS and IGLS estimators is not straightforward. Should we know the true value $\boldsymbol{\Sigma}_{\mathbf{L}}$, it is clear that the GLS estimator would be unbiased when $\mathrm{E}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] = \boldsymbol{b}_{\mathbf{L}}$. However, we use instead an estimate $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$, which is a random matrix itself. Therefore, to assess when (4.100) is unbiased we would need to calculate its expectation while considering both $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ and $\hat{\boldsymbol{b}}_{\mathrm{L}}$ of random nature with their own distribution and statistical properties, which is a formidable task that we leave for future work. Regarding the covariance, the reasoning is the same. Therefore, we do not make any claims about the bias and covariance of the GLS and IGLS position estimators.

Should we still want to calculate an approximation of the covariance, we could use the $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ returned by the algorithm as the true (deterministic) value of $\mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right]$. Using (4.100) we can then approximate the covariance of the GLS and IGLS algorithms as:

$$\begin{aligned}
\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}\right] &= [\boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}}]^{-1} \boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \hat{\boldsymbol{\Sigma}}_{\mathbf{L}} \{[\boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}}]^{-1} \boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1}\}^t \\
&= [\boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}}]^{-1} \boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}} [\boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}}]^{-1} \\
&= [\boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}}]^{-1}. \qquad (4.101)
\end{aligned}$$

What remains to be studied, however, is whether (4.101) is a good approximation in practice.

Practical Implementation

The GLS position estimator as given by Equation (4.100) depends on the calculation of two inverses: $\hat{\boldsymbol{\Sigma}}_{\text{L}}^{-1}$ and $[\boldsymbol{A}_{\text{L}}^t \hat{\boldsymbol{\Sigma}}_{\text{L}}^{-1} \boldsymbol{A}_{\text{L}}]^{-1}$. In practice, it is desirable to avoid as much as possible to calculate directly inverses of matrices. In our case it is possible to avoid completely the direct inversion of any matrices in the following manner:

The GLS estimator (4.100) is the solution of

$$\boldsymbol{S}\boldsymbol{A}_{\text{L}}\vec{\boldsymbol{s}} = \boldsymbol{S}\hat{\boldsymbol{b}}_{\text{L}}, \tag{4.102}$$

where $\boldsymbol{S}$ is the upper triangular Cholesky factor of $\hat{\boldsymbol{\Sigma}}_{\text{L}}^{-1}$. Effectively, using the normal equations we have $(\boldsymbol{S}\boldsymbol{A}_{\text{L}})^t \boldsymbol{S}\boldsymbol{A}_{\text{L}}\vec{\boldsymbol{s}} = (\boldsymbol{S}\boldsymbol{A}_{\text{L}})^t \boldsymbol{S}\hat{\boldsymbol{b}}_{\text{L}}$, from which

$$\begin{aligned}\hat{\vec{\boldsymbol{s}}} &= [(\boldsymbol{S}\boldsymbol{A}_{\text{L}})^t \boldsymbol{S}\boldsymbol{A}_{\text{L}}]^{-1} (\boldsymbol{S}\boldsymbol{A}_{\text{L}})^t \boldsymbol{S}\hat{\boldsymbol{b}}_{\text{L}} \\ &= [\boldsymbol{A}_{\text{L}}^t \boldsymbol{S}^t \boldsymbol{S}\boldsymbol{A}_{\text{L}}]^{-1} \boldsymbol{A}_{\text{L}}^t \boldsymbol{S}^t \boldsymbol{S}\hat{\boldsymbol{b}}_{\text{L}} \\ &= [\boldsymbol{A}_{\text{L}}^t \hat{\boldsymbol{\Sigma}}_{\text{L}}^{-1} \boldsymbol{A}_{\text{L}}]^{-1} \boldsymbol{A}_{\text{L}}^t \hat{\boldsymbol{\Sigma}}_{\text{L}}^{-1} \hat{\boldsymbol{b}}_{\text{L}};\end{aligned}$$

Let $\boldsymbol{R}$ be the upper triangular Cholesky factor of $\hat{\boldsymbol{\Sigma}}_{\text{L}}$, that is, $\hat{\boldsymbol{\Sigma}}_{\text{L}} = \boldsymbol{R}^t\boldsymbol{R}$. Thus, $\hat{\boldsymbol{\Sigma}}_{\text{L}}^{-1} = (\boldsymbol{R}^t\boldsymbol{R})^{-1} = \boldsymbol{R}^{-1}\boldsymbol{R}^{-t} = \boldsymbol{S}^t\boldsymbol{S}$, from which we can identify $\boldsymbol{S} = \boldsymbol{R}^{-\boldsymbol{t}}$. Substituting $\boldsymbol{S}$ into equation (4.102) we get

$$\boldsymbol{R}^{-t}\boldsymbol{A}_{\text{L}}\vec{\boldsymbol{s}} = \boldsymbol{R}^{-t}\hat{\boldsymbol{b}}_{\text{L}}.$$

The factor $\boldsymbol{R}^{-t}\hat{\boldsymbol{b}}_{\text{L}}$ is the solution of $\boldsymbol{R}^t\boldsymbol{x} = \hat{\boldsymbol{b}}_{\text{L}}$ for $\boldsymbol{x}$, which can be solved efficiently using forward substitution. The factor $\boldsymbol{R}^{-t}\boldsymbol{A}_{\text{L}}$ is the solution of $\boldsymbol{R}^t\boldsymbol{T} = \boldsymbol{A}_{\text{L}}$ for $\boldsymbol{T}$, which can be solved using any triangularization method. Finally, the solution of (4.102) can then be obtained as the solution of $\boldsymbol{T}\vec{\boldsymbol{s}} = \boldsymbol{x}$, which can be done using a triangularization technique. Putting all together and exploiting the capabilities of the backslash operator of Octave/Matlab[6], this can be done as

$$\begin{aligned}\boldsymbol{R} &= \text{chol}(\hat{\boldsymbol{\Sigma}}_{\text{L}}); \\ \hat{\vec{\boldsymbol{s}}} &= (\boldsymbol{R}^t \setminus \boldsymbol{A}_{\text{L}}) \setminus (\boldsymbol{R}^t \setminus \hat{\boldsymbol{b}}_{\text{L}}).\end{aligned}$$

Additionally, although by definition $\hat{\boldsymbol{\Sigma}}_{\text{L}}$ is positive definite, due to round-off errors in some rare cases it might result in a matrix with some very small negative eigenvalues, in which case the Cholesky decomposition will fail. In

[6]Given a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, in Octave/Matlab the solution can be found as $\boldsymbol{x} = \boldsymbol{A} \setminus \boldsymbol{b}$. The backslash operator "\" solves a linear system using different methods depending on the characteristics of the matrix $\boldsymbol{A}$, aiming at optimizing numerical stability and reducing computational complexity. In particular, it automatically applies forward/backward substitution when $\boldsymbol{A}$ is lower or upper triangular, respectively.

order to minimize this effect we exploit the structure of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$. As we can see from the formulas of its definition, $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ has equal terms in the non diagonal elements, corresponding to the term $\hat{d}_i^2 \mathbf{1}_{\mathbf{L}} \mathbf{1}_{\mathbf{L}}^t$ (see (4.94),(4.96) and (4.98)). Recall that d_{i} is the estimated distance from the node to the pivot beacon. Therefore, if we choose as pivot the beacon with the shortest distance estimate, the difference between the diagonal and non-diagonal terms in $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ will increase, which will subsequently increase also its eigenvalues and make the Cholesky decomposition more robust.

In the IGLS algorithm we used $\vec{\boldsymbol{\delta}}_n^t \mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] \vec{\boldsymbol{\delta}}_n < \epsilon$ as convergence criterion with $\epsilon = 0.01$, where $\vec{\boldsymbol{\delta}}_n$ is the increment of the position estimate between iterations, that is, $\vec{\boldsymbol{\delta}}_n = \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} - \hat{\vec{\boldsymbol{s}}}_{n-1}$. We observed that usually $\|\vec{\boldsymbol{\delta}}_n\|$ converges in a few iterations, say between three to ten, although other times it takes 20 or 30. Sometimes it converges to 0, and others it stabilizes as a positive value. With respect to the covariance $\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right]$, we follow its evolution by monitoring the characteristic error $e_c = \sqrt{\mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right]\right)}$. We observed that in most cases e_c grows initially and converges to a stable value in a similar number of iterations than $\|\vec{\boldsymbol{\delta}}_n\|$. In some rare cases, however, it takes a few more iterations. The convergence criteria chosen makes it possible to stabilize of both $\vec{\boldsymbol{\delta}}_n$ and $\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right]$ before the algorithm stops. Should the algorithm reach the last iteration, we choose as solution the position estimate at the iteration that shows the smallest value for $\vec{\boldsymbol{\delta}}^t \mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}+1}\right] \vec{\boldsymbol{\delta}}$.

4.5.3 Comparison

So far in this section, we have studied three different linear localization methods that rely on RSS-based distance estimators, each one with three variants: OLS, GLS and IGLS. Whereas the OLS estimator does not use $\mathbf{\Sigma}_{\mathbf{L}}$, the GLS and IGLS estimator do. Because $\mathbf{\Sigma}_{\mathbf{L}}$ is not known, the GLS and IGLS estimators approximate it using estimates of distances or squared distances, which adds an extra factor of uncertainty that affects the position estimates. The effects of approximating $\mathbf{\Sigma}_{\mathbf{L}}$ by its estimate could be investigated by calculating analytically the bias and covariance of (4.100) considering $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ as a random variable, which is a formidable task. Therefore, once more we aim to assess and compare the performance of the different algorithms using simulations.

Comparison of RSS methods

The simulation scenario used here is exactly the one explained already in subsection 4.4.3, which was used to compare the NLS methods. The position of the

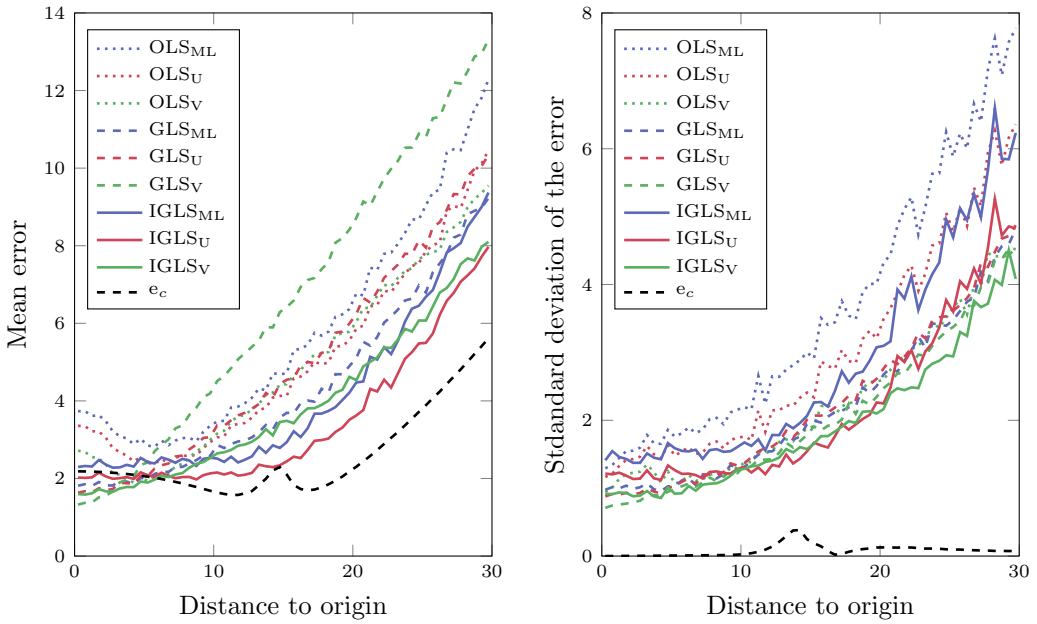
beacons, target nodes and the generated measurement vectors are the same. We also present the results for two levels of noise power: $\sigma = 2$ and $\sigma = 5$.

Figure 4.9 shows the mean error (left) and standard deviation (right) of the errors (including $e_c = \sqrt{\mathrm{Tr}(\mathrm{CRLB})}$) as a function of the annulus radius, which is approximately the distance of the target node's position to the origin of coordinates (see Panel 4.4). In all the figures, the subindices ML, U and V of the legends refer to using the ML, unbiased and LMMSE distance estimators, respectively.

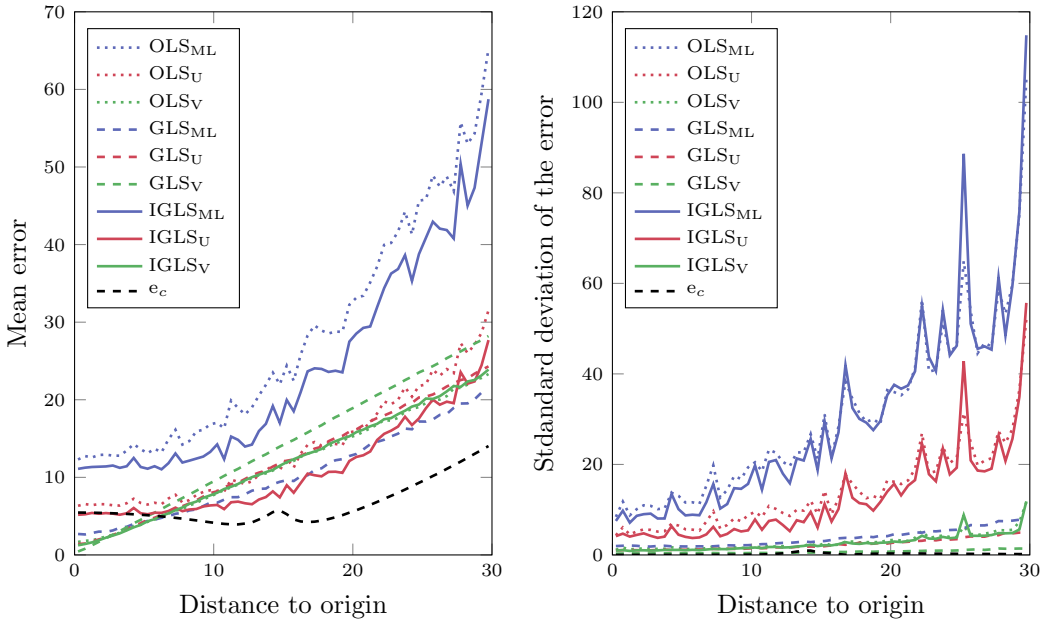
Figure 4.10 shows the aggregated comparison of all the methods. The horizontal axis presents the mean error of all the position estimates and the vertical axis its standard deviation. Again, the better the algorithm performed globally in our simulations the closer its corresponding marker will be to the lower left corner of the figure.

When comparing the OLS methods, using the unbiased and LMMSE distance estimators result in better estimates compared to using ML. For a moderate noise power, the difference in the mean error is minor but consistent for all the distances, and more accused for positions close to the origin of coordinates (see Panel 4.9a). The difference in the standard deviation of the error is clearer at all distances, however. The difference in performance becomes much more evident when we increase the noise power (see Panel 4.9b). Note how the $\mathrm{OLS_V}$ falls well below the characteristic error when the node is close to the origin of coordinates. The $\mathrm{OLS_U}$ performs clearly better than the $\mathrm{OLS_{ML}}$ for large noise, both in terms of the mean and standard deviation of the error, but the dispersion of the error is still significantly larger than with the $\mathrm{OLS_V}$.

The global difference in performance among the OLS methods can be seen clearly in Figure 4.10. As we can see, the plus signs are clearly separated for both $\sigma = 2$ and $\sigma = 5$, with the one for the $\mathrm{OLS_V}$ being the closest to the origin. Note how the distance among markers becomes significantly larger for $\sigma = 5$ with respect to the case with $\sigma = 2$.

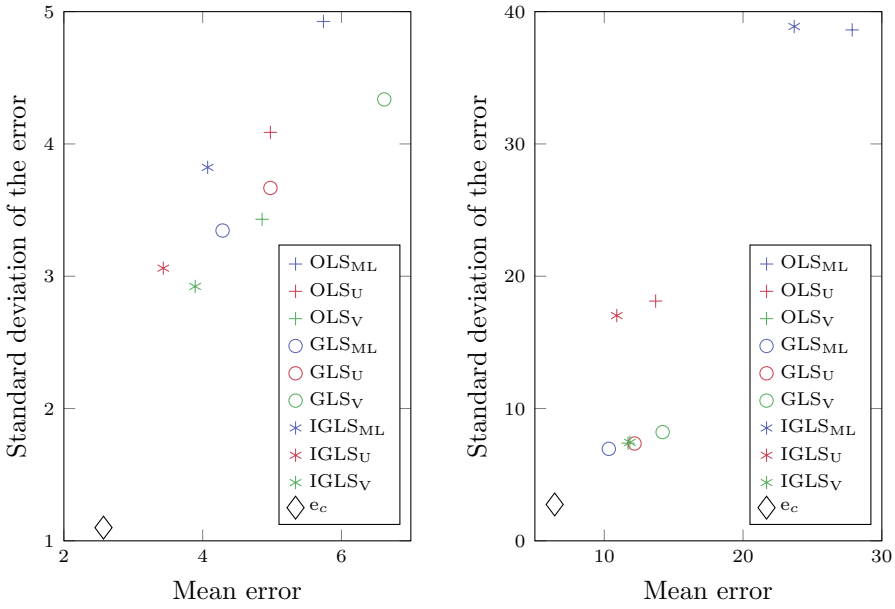


(a) Mean and standard deviation of the error for $\sigma = 2$



(b) Mean and standard deviation of the error for $\sigma = 5$

Figure 4.9: Comparison of LLS methods ($z_0 = -70$, $n = 2$).



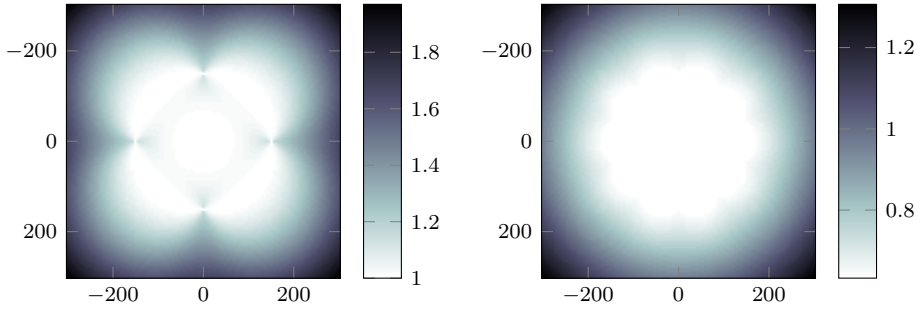
(a) $\sigma = 2$ (b) $\sigma = 5$

Figure 4.10: Global comparison of the LLS methods. Model parameters: $z_0 = -70$, $n = 2$, $\sigma = 2$ (left) and $\sigma = 5$ (right).

When comparing the GLS methods globally, surprisingly the situation is rather the opposite. The GLS_{ML} is clearly the best method for both levels of noise power in terms of the mean and standard deviation of the error, whereas the GLS_{V} is the worse (see Figure 4.10).

Regarding the differences between OLS and GLS methods, using the latter ones does not result in an increase of performance when using all the distance estimators. In fact, GLS_{v} performs worse than OLS_{v}, except for rather small distances to the origin (see Figure 4.9). The largest improvement occurs when using the ML distance estimator. Overall, we can see that the GLS_{ML} version in general offers a significant level of improvement in performance over all the OLS estimators for all distances and noise levels being tested (see Figure 4.9). On a global comparison, though, the OLS_{v} method follows closely (see Figure 4.10).

Continuing with the iteration of the IGLS methods brings in general clearly better performance with respect to the corresponding OLS and GLS methods. The exception is when using ML distance estimators with larger noise power, which results in a significant increase of the mean and variability of the error.



(a) 4 beacons. (b) 10 beacons.

Figure 4.11: Characteristic errors (σ-normalized) of the simulation settings.

Which method should be used in a general scenario? The OLS estimators are clearly the least computationally demanding. Should we choose an OLS method, it performs best when using LMMSE distance estimators, especially in the presence of large levels of noise. GLS methods require in addition the Cholesky decomposition of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$, and therefore are more computationally costly. Among them, using ML distance estimators seems a good choice, and brings a consistent performance improvement over any of the OLS methods. Going further for an IGLS method can increase further the performance, but at a higher computational cost. Among the IGLS estimators the GLS_{U} results in smallest average errors in general, although the GLS_{V} gives smaller error variability.

One important aspect to note is that the results of the OLS methods showed a significant dependency on the beacon selected as pivot. Choosing different pivots could lead to changes in mean errors of a factor of two, and thus their curves in the previous figures could have shown clearly larger errors than any GLS method. Although our choice of pivot does not guarantee the minimum error in all circumstances, in practice it was observed to be rather good, in the sense that it gave usually the smallest errors when compared to almost all other choices tested in practice. However, the GLS and IGLS were not affected in any noticeable way. So, yet another advantage of these methods is their immunity to the selection of pivot.

Comparison of AGN Model Based Methods

We proceed now to compare using simulations the performance of the different linear localization methods under the AGN model. For convenience we will also present the results of the NLS approach from Subsection 4.4.1, which we solved using the Taylor-series linearization method (Newton-Raphson).

The simulation scenario and methodology used to calculate errors and plot results is also the one explained already in Subsection 4.4.3, but with the beacons placed in a circle with a 150 m radius. We present results when using four and ten beacons. Figure 4.11 shows their respective σ-normalized characteristic error as a function of the 2D position.

Figure 4.12 shows the average and standard deviation of the errors for the different estimators as a function of the annulus radius (aprox. node distance to the origin) up to 300 m. Figure 4.13 presents the same results for a larger scale with radius up to 2 km. Figure 4.14 shows the mean vs. the standard deviation of the errors for the different algorithms for the case of using up to 300 m radius.

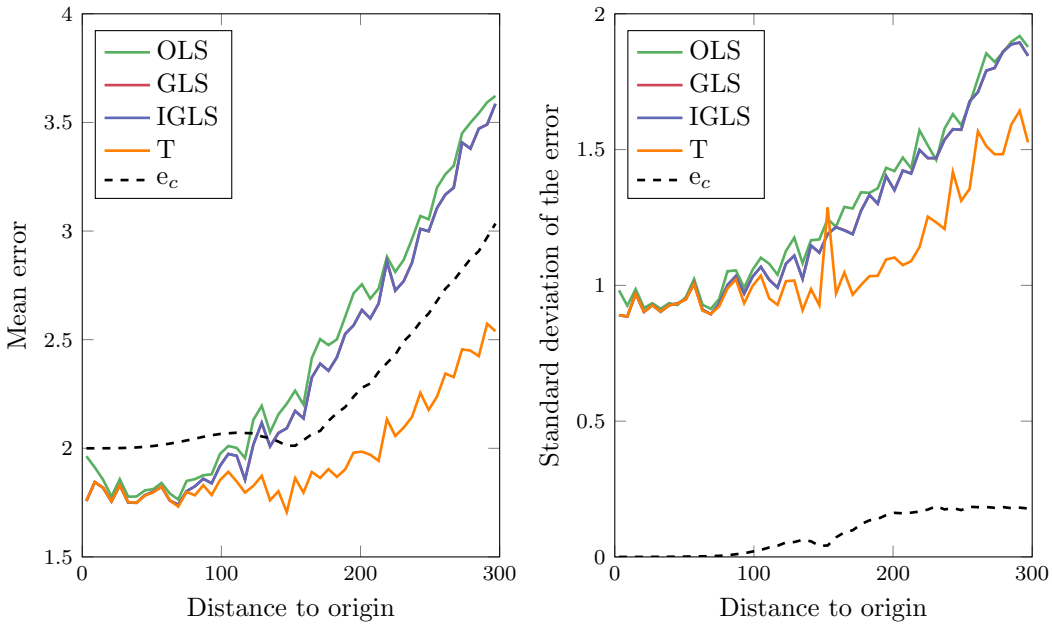
Similarly as with the log-normal model, the selection of the pivot affects the results of the OLS significantly, whereas GLS and IGLS are essentially not affected. The figures presented here are also created using the closest beacon to the target node as the pivot, using the estimated distances as measure of closeness. Again, this choice gave in general the best results. Had we chosen other pivot, the difference between the OLS and GLS would be significantly larger in all cases.

In contrast to the log-normal model, the average and standard deviation of the GLS and IGLS are indistinguishable from each other. The IGLS iteration does not provide any extra performance with respect the GLS. This can be seen in all the figures, and especially in Figure 4.14. We observed that the IGLS algorithm changed the estimated position in the order of millimetres in the first iteration, and less in further ones.

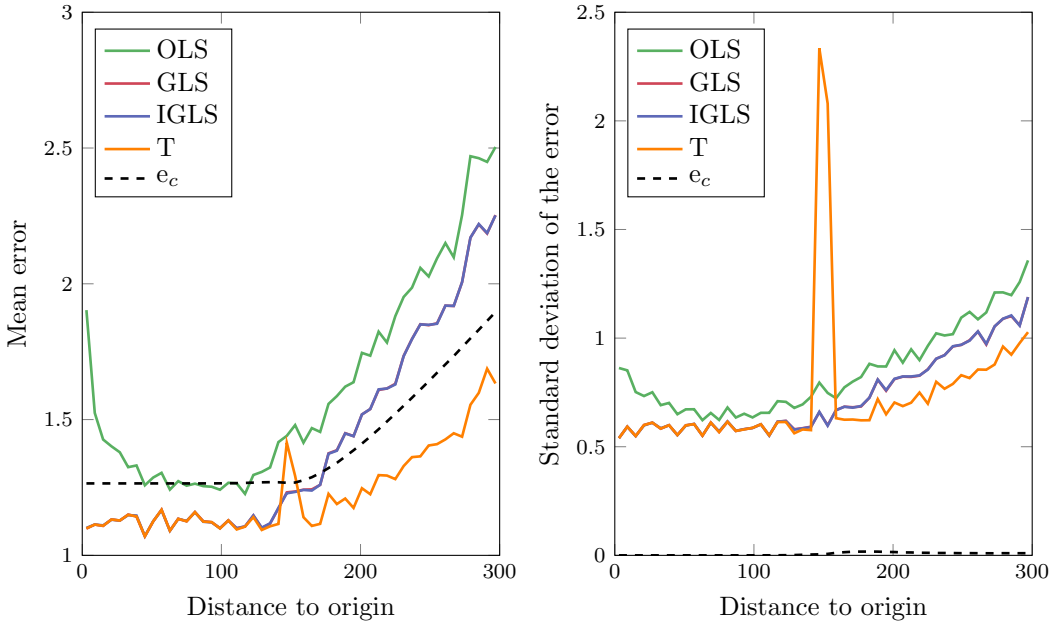
In general, there are two areas with clearly different behaviour of the errors separated by the circle with 150 m radius, where the beacons are located. We refer to the radius of this circle as d_b. In the inner area (radius $\leq d_b$) the characteristic error remains more or less constant, the same as the error of the Taylor method. In the outer area (radius $> d_b$) both errors increase with the distance more or less linearly, but with slightly different slope. This difference is better observed in Figure 4.13, which presents results for larger distances to the origin.

In the inner area, the mean and standard deviation of the error of the GLS and Taylor methods are the same until a breakpoint, and do not change much with the distance to the origin. The breakpoint is situated at a distance smaller than d_b. After the breakpoint the errors of the linear methods begin to increase, whereas the error of the Taylor method remains stable until d_b. The mean and standard deviation of the OLS estimator before the breakpoint have a similar behaviour as the GLS when using four beacons, but with a slightly larger mean and average error (see Figure 4.12a).

For radii larger than d_b, the errors associated to all the methods increase linearly, but with different slopes. The slope for the OLS method is larger than for the GLS methods, while the Taylor method has the smallest slope. This can be seen more clearly in Figure 4.13, in which the largest radius is 2 km.

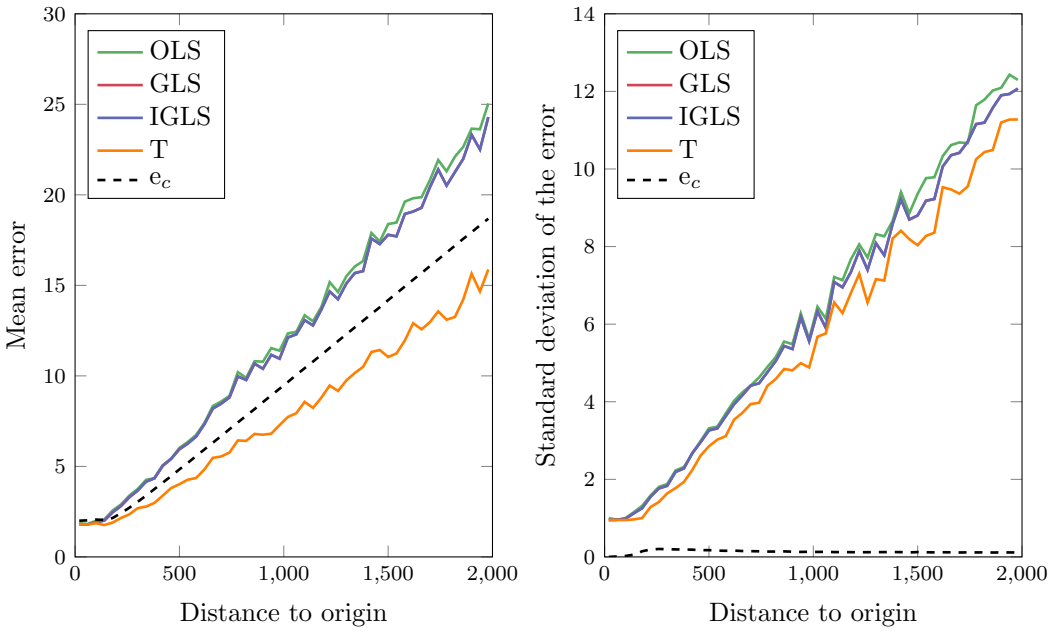


(a) 4 beacons.

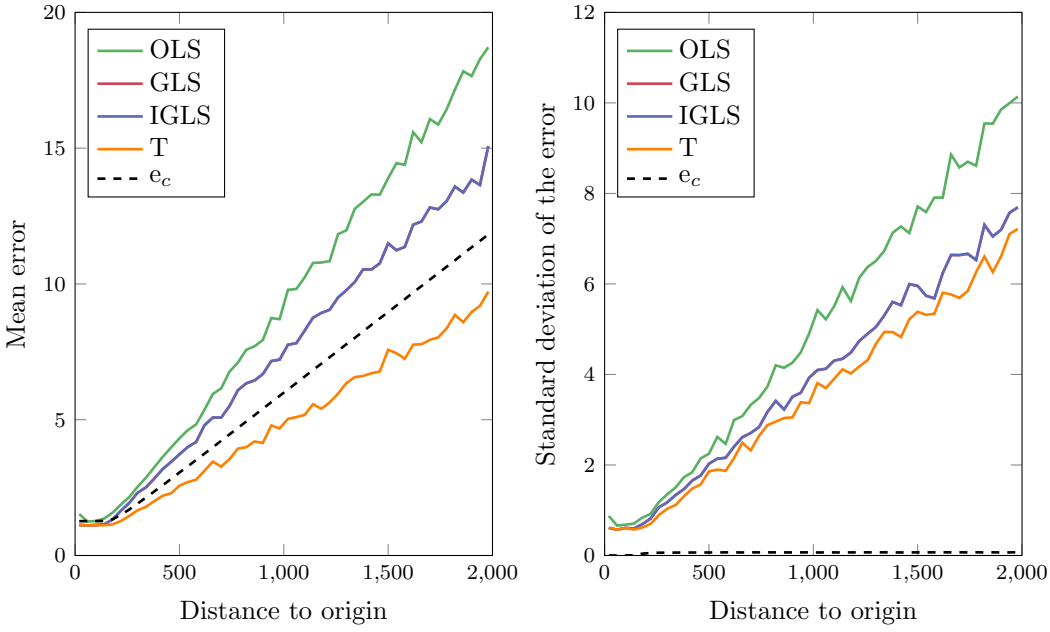


(b) 10 beacons.

Figure 4.12: Mean and standard deviation of the error when using 4 and 10 beacons ($\sigma = 2$).

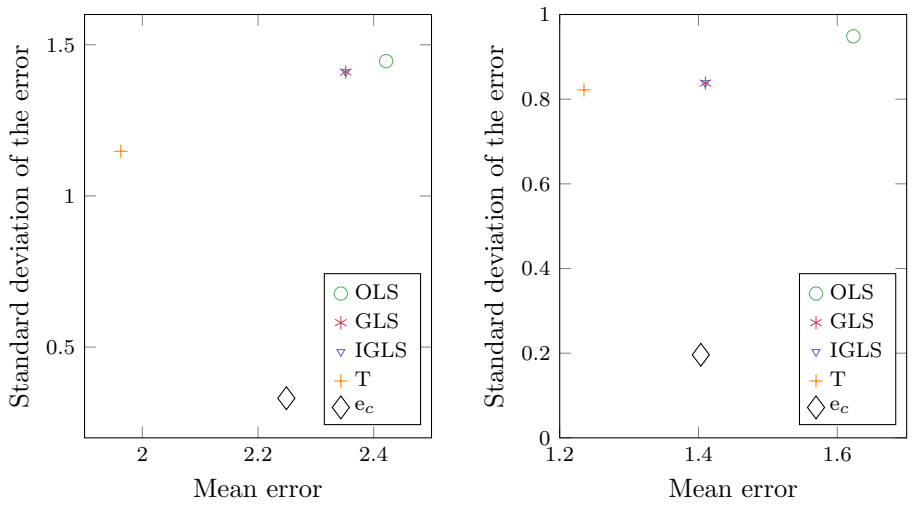


(a) 4 beacons.



(b) 10 beacons.

Figure 4.13: Comparison among LLS methods under the AGN model ($\sigma = 2$).



(a) 4 beacons.

(b) 10 beacons.

Figure 4.14: Global comparison of the LLS methods using AGN models for the simulations with maximum radius of 300 m ($\sigma = 2$). The markers show the mean vs. the standard deviation of the errors when using their associated algorithm.

Adding more beacons has clear benefits. First, the difference between the OLS and GLS methods become more accentuated. Also, the breakpoint becomes closer to d_b as the number of beacons increases, causing the GLS perform more similar to the Taylor method in the inner area (compare panels 4.12a and 4.12b). In the outer area, the slopes of the lines of the error's mean and standard deviation naturally decrease as the number of beacons increases. With four beacons the lines corresponding to the GLS are rather close to the line associated with the OLS method (see Figure 4.13a). However, the lines for the GLS move closer to the line of the Taylor method as the number of beacons increases (see 4.13b).

Had we used only three beacons in the simulations, the results of all the linear methods (OLS, GLS and IGLS) would have been exactly the same. This is because the algebraic transformation required for the analytical linearization reduces the number of equations in one. Having three beacons leaves us with two equations, and we have two coordinates to estimate. The system of equations then becomes determinate. Therefore, the performance of the GLS algorithms will be better than that of the OLS when using four beacons or more for 2D localization.

The effect of changing σ is only to change the mean and standard deviation of the errors proportionally.

4.6 Maximum Likelihood

Until now all the localization methods studied were based on the LSE criterion. In this section we describe the principle of maximum likelihood for the localization problem, which will be applied later in this thesis.

Suppose we have a probabilistic model for the observations z, $p(z;\boldsymbol{\theta})$, which depends on a vector of L parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_L)$. The likelihood of one measurement z_k is then $p(z_k;\boldsymbol{\theta})$. Assuming IID samples, the likelihood of a set of observations $\boldsymbol{z}$ is the product of the individual likelihoods, that is:

$$p(\boldsymbol{z};\boldsymbol{\theta}) = \prod_{k=1}^{K} p(z_k;\boldsymbol{\theta}). \tag{4.103}$$

The principle of maximum likelihood is simple: maximize the likelihood of the observations conditioned on the parameters that we want to estimate. The estimate is the value of the parameter that maximizes the likelihood:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{z};\boldsymbol{\theta}). \tag{4.104}$$

Instead of maximizing (4.104) directly, usually it is easier to maximize the log-likelihood

$$L(\boldsymbol{\theta}) \equiv \ln p(\boldsymbol{z};\boldsymbol{\theta}) = \ln \prod_{k=1}^{K} p(z_k;\boldsymbol{\theta}) = \sum_{k=1}^{K} \ln p(z_k;\boldsymbol{\theta}). \tag{4.105}$$

Being the logarithm a monotonic transformation, the value of $\boldsymbol{\theta}$ that maximizes the likelihood is the same as the one that maximizes the log-likelihood.

A necessary condition for finding find a maximum of the log-likelihood is that its gradient must be equal to zero. From (4.105), the derivative of the log-likelihood with respect to the l-th parameter is

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \theta_l} = \sum_{k=1}^{K} \frac{\partial}{\partial \theta_l} \ln p(z_k;\boldsymbol{\theta}). \tag{4.106}$$

As we can see, each measurement z_k contributes to the partial derivative with respect to θ_l with the partial derivative of its log-likelihood. We can then form the matrix of the contribution to the gradients as

$$\boldsymbol{J}(\boldsymbol{\theta}) \equiv \begin{pmatrix} \frac{\partial \ln p(z_1;\boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial \ln p(z_1;\boldsymbol{\theta})}{\partial \theta_L} \\ \frac{\partial \ln p(z_2;\boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial \ln p(z_2;\boldsymbol{\theta})}{\partial \theta_L} \\ \vdots & \ddots & \vdots \\ \frac{\partial \ln p(z_K;\boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial \ln p(z_K;\boldsymbol{\theta})}{\partial \theta_L} \end{pmatrix} = \begin{pmatrix} \boldsymbol{J}_1(\boldsymbol{\theta}) \\ \boldsymbol{J}_2(\boldsymbol{\theta}) \\ \cdots \\ \boldsymbol{J}_K(\boldsymbol{\theta}) \end{pmatrix},$$

which is effectively the $K \times L$ Jacobian of the log-likelihood when the log-likelihood is interpreted as depending on the K individual likelihoods, $p(z_k; \boldsymbol{\theta})$. $\boldsymbol{J}_k(\boldsymbol{\theta})$ is the vector of contributions to the gradient corresponding to the k-th sample. The gradient is then the 1xL vector resulting of adding up the rows of the Jacobian:

$$\frac{\partial L}{\partial \boldsymbol{\theta}} = \sum_{k=1}^{K} \big[J_{k1} \ , \ \dots \ , \ J_{kL} \big].$$

The points $\boldsymbol{\theta}$ for which the gradient is zero are the stationary points of the log-likelihood. A necessary condition to ensure that a stationary point corresponds to a maximum is that the Hessian is negative definite. The Hessian is the $L \times L$ matrix of the second partial derivatives

$$\boldsymbol{H}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial^2 \ln p(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_1^2} & \cdots & \frac{\partial^2 \ln p(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_L} \\ \frac{\partial^2 \ln p(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_1} & \cdots & \frac{\partial^2 \ln p(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_2 \partial \theta_L} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \ln p(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_L \partial \theta_1} & \cdots & \frac{\partial^2 \ln p(\boldsymbol{z};\boldsymbol{\theta})}{\partial \theta_L^2} \end{pmatrix}.$$

Finite sample ML estimators can be biased and do not have any optimality properties. However, under certain regularity conditions, [7] it can be shown that they are asymptotically consistent, efficient and normally distributed. This means that, as the number of samples grows, the estimator converges to the real value of the parameter being estimated, its covariance approaches the minimum possible covariance for any unbiased estimator and that the estimator is normally distributed. These properties are only valid when the model is not misspecified (see [Kay, 1993] or [Davidson and MacKinnon, 2004]).

The covariance of ML estimators can be approximated using different methods [Davidson and MacKinnon, 2004, p.415]. One can use the covariance of the asymptotic distribution of the estimator. This covariance matrix is the inverse of the information matrix, which is defined as the expectation of the covariance of the gradient

$$\boldsymbol{I}(\boldsymbol{\theta}) = \mathrm{E}\left[\frac{\partial L}{\partial \boldsymbol{\theta}} \frac{\partial L}{\partial \boldsymbol{\theta}}^t \right], \tag{4.107}$$

where the expectations are computed with respect $\boldsymbol{z}$. The FIM estimator is then

$$\widehat{\mathrm{Cov}}\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{I}^{-1}(\hat{\boldsymbol{\theta}}),$$

[7] The regularity conditions, in short, are the necessary conditions that allow us to expand the log-likelihood in a Taylor series with respect $\boldsymbol{\theta}$ and truncate it, and to calculate the expectations of the gradient and Hessian. [Greene, 2011, ch. 14.4].

which is effectively the CRLB. Note that it does not depend on the observations, $\boldsymbol{z}$, as its dependency has been removed when calculating the expectation in (4.107). Another possibility is to use the so-called empirical Hessian estimator

$$\widehat{\mathrm{Cov}}\left[\hat{\boldsymbol{\theta}}\right] = -\boldsymbol{H}^{-1}(\hat{\boldsymbol{\theta}}),$$

which requires the ability to calculate the Hessian of the log-likelihood, and does depend on $\boldsymbol{z}$. Another option is to use the outer product of the gradient

$$\widehat{\mathrm{Cov}}\left[\hat{\boldsymbol{\theta}}\right] = \left[\boldsymbol{J}^t(\hat{\boldsymbol{\theta}})\boldsymbol{J}(\hat{\boldsymbol{\theta}})\right]^{-1}.$$

Finally, we also have the sandwich estimator:

$$\widehat{\mathrm{Cov}}\left[\hat{\boldsymbol{\theta}}\right] = \boldsymbol{H}^{-1}(\hat{\boldsymbol{\theta}})\boldsymbol{J}^t(\hat{\boldsymbol{\theta}})\boldsymbol{J}(\hat{\boldsymbol{\theta}})\boldsymbol{H}(\hat{\boldsymbol{\theta}}).$$

In essence, the amount of information available is related to the curvature of the log-likelihood around the estimate, and the previous covariance estimators are measures of that. The larger the curvature, the more the information is available. In this thesis we will use the empirical Hessian estimator.

In case of measurement models with zero mean normally distributed errors, the likelihood of one observation is

$$p(z_{\mathrm{k}};\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(z_{\mathrm{k}}-\bar{z}_{\mathrm{k}})^2}{2\sigma_k^2}\right], \tag{4.108}$$

and the log-likelihood of a set of K IID measurements $\boldsymbol{z} = [z_1, z_2, ..., z_{\mathrm{K}}]$ becomes

$$\begin{aligned} L(\boldsymbol{\theta}) &= -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=1}^{K}(z_{\mathrm{k}}-\bar{z}_{\mathrm{k}})^2 \\ &= -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{z}-\bar{\boldsymbol{z}})^t(\boldsymbol{z}-\bar{\boldsymbol{z}}), \end{aligned} \tag{4.109}$$

where $\bar{\boldsymbol{z}} = [\bar{z}_1, \bar{z}_2, ..., \bar{z}_{\mathrm{K}}]$. Clearly, maximizing (4.109) requires minimization of the factor $(\boldsymbol{z}-\bar{\boldsymbol{z}})^t(\boldsymbol{z}-\bar{\boldsymbol{z}})$, which is the sum of squared errors. This criterion is the same as the least squares criterion, and thus the solution is also the same.

The AGN and log-normal models of (4.4) and (4.3) have normally distributed errors. The equivalence between the LS and ML criterion then implies that the results regarding the non-linear estimators presented in Section 4.4 are also applicable to ML estimators.

In either case, the resulting problem is non-linear on the position to be estimated, and therefore we have to rely on numerical methods to find the solution.

In Section 4.4 we used a Newton-Raphson solver, which was convenient for different theoretical calculations within this chapter. In practice, however, there are other methods with better convergence properties. In this thesis we will use the damped Broyden–Fletcher–Goldfarb–Shanno (BFGS) solver ([Nocedal and Wright, 2006, ch. 6]) for ML estimation, and its large-scale bounded variation bound-constrained large-scale BFGS (L-BFGS-B) from [Byrd et al., 1995].

4.7 Discussion

In this chapter we have studied many localization techniques based on the LSE optimization criterion. The goal of this section is to recapitulate the relevant conclusions of the chapter and to present a global comparison of the best methods.

In general, the LSE criterion leads to non-linear problems in the position that we want to estimate, and therefore we have to rely on iterative methods to solve the problem. One such iterative method is the Newton-Raphson method, sometimes known as Taylor-series linearization estimation, which we used in Section 4.4 to compare different localization schemes.

In subsection 4.4.1, we formulated the position estimation problem as a minimization of the sum of squared distance errors, for which we need distance estimates. When using the AGN model we used as distance estimates the measurements themselves, which we proved in Subsection 4.2.1 to be the efficient choice. When using RSS under the log-normal model, we saw that there is no efficient distance estimator, and we explored the possibility of using three different estimators. However, it is also possible to pose the problem as a minimization of the sum of squared RSS errors rather than converting RSS to distance estimates first and then minimizing in the distance domain. We saw using simulations that staying in the RSS domain yields in general better results, with the exception of cases with high noise power and very good geometrical configuration of the problem, cases in which an optimization in the distance domain using the LMMSE distance estimators can provide a smaller average error. But for a general case, an optimization in the RSS domain proved to be superior, and therefore we select it as the best solution among the NLS methods.

In Section 4.5 we used an algebraic transformation of the original equations that turns them into a linear system. This opened the door to using three different families of solvers: OLS, GLS and IGLS. These require either distance or squared distance estimates, which can be obtained using the estimators studied in Section 4.2. We then compared all of the linear LS methods, and assessed the improvements introduced by the newly proposed GLS and IGLS families.

Finally, in Section 4.6 we posed the problem using ML optimization, and argued that the solution is necessarily the same as with the NLS problem proposition when the LSE criterion penalizes directly the observations.

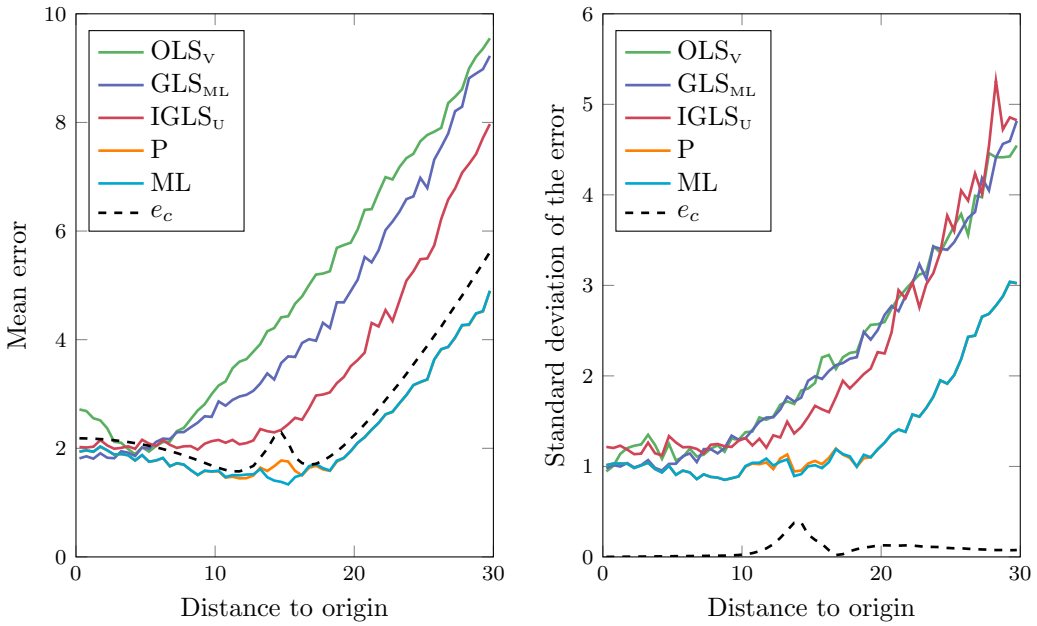
Figures 4.15 and 4.16 reproduce the results of the best methods from our simulations for an easier comparison. The legend ML is used to represent the localization method using the ML criteria and the BFGS solver, while the legend P is used to indicate the method using the LSE criteria with a Taylor solver in the RSS domain. As expected, the P and ML methods show the same results in all cases except in positions close to the circle with a 15 m radius, cases in which the Taylor method has more convergence problems.

The ML and Taylor methods generally perform better than the rest both in terms of mean and standard deviation of the error and for the two noise levels tested (see Figure 4.16). However, in cases with a good geometrical configuration of the problem and relatively large noise, Panel 4.15b suggests that using the $\mathrm{OLS_V}$ estimator might result in less average error.

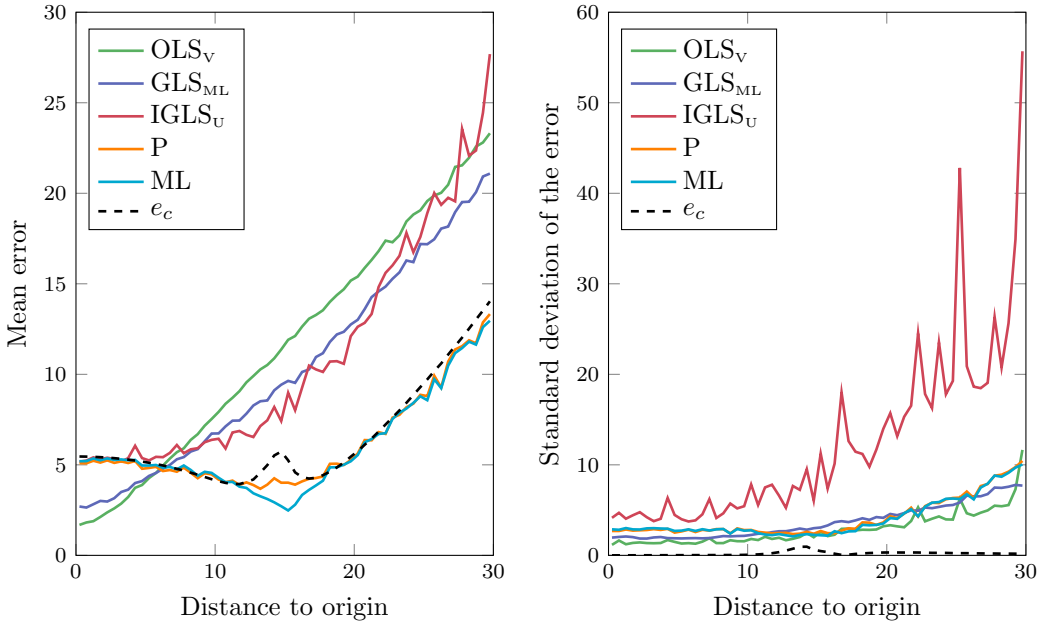
Despite the overall better performance of the ML estimator, it has two main drawbacks with respect to the linear ones: it requires an initial position estimate and, in general, is more computationally demanding. Finding an initial position estimate is not trivial. It must be *close enough* to the true position; otherwise, the algorithm might converge to a non-global maximum or might not converge at all. In principle, one can use a grid-search with a fine enough resolution to guarantee sufficient closeness of the initial estimate. However, where to place the grid and choosing its resolution are not automatically clear issues. In any case, the extra computations of the log-likelihood required by the grid-search make the whole localization method even more computationally demanding.

On the other hand, the linear methods do not require any initial position estimate, not even the iterative IGLS family. Additionally, they can be very computationally efficient, especially when the dimensionality of the measurement vector is small.

The most extreme case is the family of OLS position estimators. From (4.86) we see that the matrix $\boldsymbol{C}_{\mathrm{L}}$ depends only on the position of the beacons and not on the measurements. Therefore, $\boldsymbol{C}_{\mathrm{L}}$ needs to be calculated only once for a given setup, and it can then be reused for all the localization processes. Furthermore, is very easy to compute, as it only requires calculating the inverse $[\boldsymbol{A}_{\mathrm{L}}^t \boldsymbol{A}_{\mathrm{L}}]^{-1}$ of dimension 2x2 plus a matrix multiplication. Once calculated $\boldsymbol{C}_{\mathrm{L}}$, estimating position in that setup requires only calculating $\hat{\boldsymbol{b}}_{\mathrm{L}}$ and a matrix multiplication. Given its simplicity, OLS methods are good candidates for very computationally constrained applications. In particular, using LMMSE distance estimators gave the best overall results in its class, which, as we have seen, can even outperform ML under very good geometrical conditions and with large measurement noise.

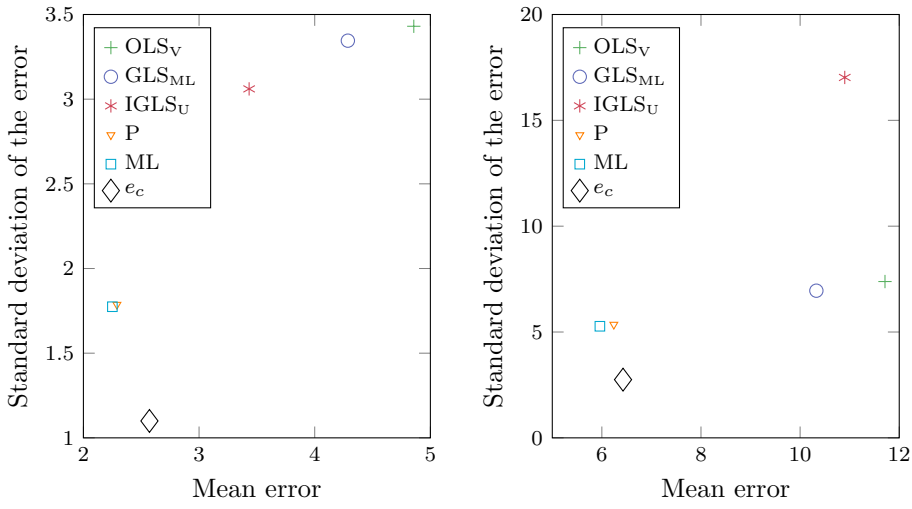


(a) Mean and standard deviation of the error for $\sigma = 2$



(b) Mean and standard deviation of the error for $\sigma = 5$

Figure 4.15: Comparison of the best methods. Model parameters: $z_0 = -70$, $n = 2$.



(a) $\sigma = 2$ (b) $\sigma = 5$

Figure 4.16: Comparison of the best methods (Model parameters: $z_0 = -70$, $n = 2$). Mean and standard deviation of the error for $\sigma = 2$ (left) and $\sigma = 5$ (right).

Should we desire better estimates, we can still rely on the GLS and IGLS families, but at a larger computational cost. Because $\hat{\hat{\boldsymbol{s}}}$ now depends on $\hat{\boldsymbol{\Sigma}}_{\text{L}}$ (see (4.100)) and $\hat{\boldsymbol{\Sigma}}_{\text{L}}$ on the distance estimates, it is not possible to pre-compute part of the estimator and reuse it as easily as with the OLS estimators. Still, the increase of performance is significant, especially with moderate noise levels (see Panel 4.15a).

In any case, the study presented in this chapter is based on analytical formulas and computer simulations, which can be rather far from reality. In particular, as we have argued in previous chapters, it is well known that the log-normal model does not necessarily explain well the RSS in all scenarios, nor is its error necessarily Gaussian. In the next chapter we will apply all the localization methods studied so far to estimate the position of the nodes in our WSN using the real data obtained during the experiments, and we will see how they perform in reality.

Chapter 5

Node Localization Under IID Noise

Contents

In Chapter 4 we studied via simulations the performance of several source localization algorithms. This chapter is dedicated fundamentally to the application of those algorithms in order to estimate the position of the nodes in our experimental setup and to find model identification strategies that can help improve the overall localization performance.

Of the studied methods, ML estimation was considered the best in a general scenario. Using a robot as a mobile beacon poses characteristic problems for ML position estimators when using a high measurement sampling rate: the appearance of a distinctive type of local maxima and overconfidence of the estimates. We begin the chapter by studying these problems and proposing solutions in Section 5.1. One of the solutions is the Maximize-Explore algorithm, which helps gradient-based solvers overcome local maxima. The underlying logic and the implementation details will be presented in Section 5.2.

Section 5.3 continues with an assessment of the performance of ML and the different families of linear position estimators studied in Chapter 4. In the light of the variability of the model parameters observed in Chapter 3, we investigate the possibility of using an individual model (IM) for each node instead of a common one (CM) for all the nodes, and we compare the localization results. Using IMs clearly yields better results in general, and this motivates us to study further the relevance of the model identification process and its impact on localization.

Traditionally, a single site-specific CM is used in WSNs by all the nodes. We have seen in Chapter 3 that models identified with different nodes can have significantly different parameter values. We will also see in Section 5.3 the benefits of using IMs. It is therefore natural to ask ourselves about the impact of using CMs in WSNs identified with a pair of nodes. How does it affect the localization error in the rest of the nodes? Is this effect relevant? And if so, can we do something about it? In order to answer these questions, in Section 5.4 we study the sensitivity of the ML position estimator to changes in the model parameters. Then in Section 5.3 we examine in detail the advantages of two model identification strategies: using data from multiple nodes when fitting a common model and using individual models instead of a common one.

The benefits associated with using IMs are due to the models' specialization: each IM now has to explain the observations of only its corresponding node. Encouraged by this result, in Section 5.6 we investigate the possibility of specializing further the IMs. So far we have used the log-normal model to predict the RSS. However, as was discussed in Section 2.6, due to its limited expressiveness it can give biased predictions in inhomogeneous environments. To alleviate the limitation due to its linearity, we investigate the possibility of using polynomials of higher degree instead of the log-distance model. Furthermore, we also experiment using mixture of Gaussians instead of normally distributed variables to model the errors.

5.1 ML and High Measurements' Linear Density

Let us rewrite here for convenience the log-likelihood function of a vector of K IID RSS measurements $\boldsymbol{z} = (z_1, z_2, ..., z_{\mathrm{K}})^t$ (see (4.109)):

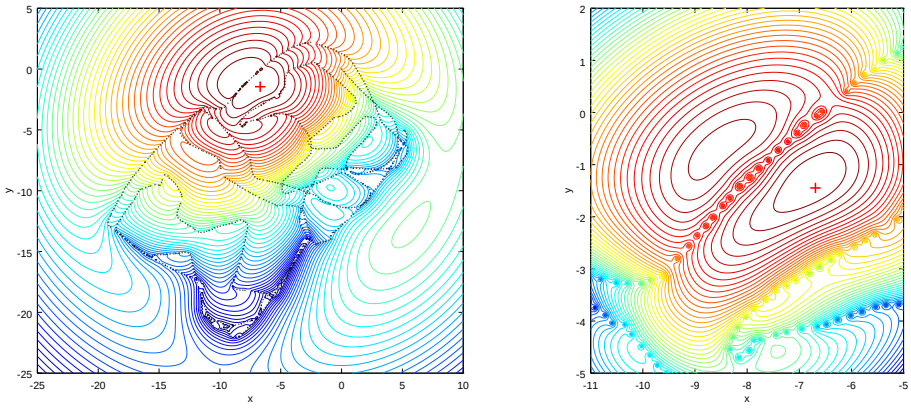
$$\begin{aligned} L(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) &= -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{k=1}^{K}(z_{\mathrm{k}} - \bar{z}_{\mathrm{k}})^2 \\ &= -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}(\boldsymbol{z} - \bar{\boldsymbol{z}})^t(\boldsymbol{z} - \bar{\boldsymbol{z}}), \end{aligned} \tag{5.1}$$

where

$$\bar{z}_{\mathrm{k}} = z_0 - 10\, n \log\left(d_{\mathrm{k}}/d_0\right). \tag{5.2}$$

Being that $L(\vec{\boldsymbol{s}}, \boldsymbol{\theta})$ in non-linear on the node's position, finding its maximum requires numerical iterative methods that start from an initial guess. Gradient-based methods can become trapped in local maxima, and the relevance of this complication depends on the shape of the log-likelihood function and the position given as initial guess. The shape of the log-likelihood function depends on the position of the beacons. As we can see from (5.2), the value of $\bar{z}_{\mathrm{k}}$ tends to ∞ as the distance $d_{\mathrm{k}} = ||\vec{\boldsymbol{s}} - \vec{\boldsymbol{r}}_{\mathrm{k}}||$ approaches zero. This implies that the log-likelihood function (5.1) has an asymptote in each of the positions where the beacons are located, $\vec{\boldsymbol{r}}_{\mathrm{k}}$, with values tending to $-\infty$ as $\vec{\boldsymbol{s}}$ approaches any $\vec{\boldsymbol{r}}_{\mathrm{k}}$, and it will be large and negative for positions in the near surroundings. Because of this, the problem associated with the presence of local maxima acquires special relevance in the context of this thesis.

When using a robot as a mobile beacon, and when collecting a high density of measurements along its trajectory, the asymptotes introduced by the beacons are close to each other. Thus, the log-likelihood will exhibit large negative values and with strong fluctuations when evaluated in different positions along the trajectory of the robot, depending on how close these position are to any of the measurement points (beacons). Gradient-based methods might then find difficulties passing over this curve. The result is that there can be areas with local maxima created by the robot trajectory in which the solver can become trapped. We can see this clearly in Panel 5.1a, which shows the contour of the log-likelihood function of the data collected by one of the nodes in the basketball field. As we can see, there is a general tendency of the likelihood to increase towards a global maxima. But the contour lines have a tendency to be parallel to the trajectory of the robot at points close to the trajectory. This creates many local maxima. Panel 5.1a has been generated by first calculating the log-likelihood in a dense grid of points and then calculating and plotting 50 contour lines. Had we plotted more contour lines, we would see more contour lines parallel to the trajectory of the robot.



(a) Complete area.

(b) Zoomed area.

Figure 5.1: Log-likelihood contours for a sample robot trajectory in the basketball field. The global maxima is marked with a '+' sign. In Panel 5.1a the measurement points are marked with a black dot.

Panel 5.1b is a detailed contour in an area close to the global maximum, in which we can see more clearly this phenomena. Note the asymptotes corresponding to each of the measurement points, and how the contour lines get parallel to the trajectory of the robot.

To remedy this problem we will explore the following solutions.

1. The first option is the classic approach of using a grid search (GS) to find a good enough initial position estimate with which to start a gradient-based solver.

2. Alternatively, we can start the solver at any point and let it converge to whatever maximum. We can then explore the area close to the solution trying to determine whether the maximum is local or not, and, if so, try to escape from it. The result is an algorithm that we call Maximize-Explore, which will be explained in Section 5.2.

3. Finally, we can reduce the number of measurement points. Choosing them distant enough from one another will separate the asymptotes in the log-likelihood function, thereby reducing the risk of appearance of local maxima due to the high linear density of measurements. This approach will be studied later in Chapter 6, when we introduce autocorrelation models.

Should we try the first approach, the GS can guarantee the convergence of a solver to a global maximum in a specified domain as long as the grid is fine

enough. However, it has some drawbacks. First, we have to know *a priori* the area in which to do the search, which can be quite large. Second, we do not know *a priori* the position of the local maxima nor how close they are to each other, and likewise how fine the grid should be in order to avoid them. Clearly the finer the grid the more likely the local maxima will be evaded. Third, the number of function evaluations required by a grid search grows proportionally to the area being explored and squared to the resolution used, making GS a computationally expensive algorithm. This, in turn, imposes limitations on the size of the area that can be explored and the resolution of the grid. The second approach (the Maximize-Explore algorithm) basically aims to decrease the computational requirements of plain GS by making the search more efficient.

There is still another problem related to using a high number of measurements and the IID errors assumption. This assumption grants that every sample adds exactly the same amount of information, independently of the number and how close the samples are collected to each other. When using a large number of measurements this will result in a very peaky log-likelihood functions, which in turn will lead to over-confident position estimates. To solve this problem, in this chapter[1] we substitute the likelihood of a single measurement $p(z|\vec{\boldsymbol{s}}, \boldsymbol{\theta})$ by $p(z|\vec{\boldsymbol{s}}, \boldsymbol{\theta})^\alpha$, with $0 < \alpha < 1$. This reduces the information that each measurement provides, making the likelihood more flat. This technique of over-confidence reduction is used, for example, in beam models of range finders [Thrun et al., 2005, p. 167] and RSS localization [Ferris et al., 2006]. The flattened log-likelihood function then becomes

$$L(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = -\alpha \frac{K}{2} \ln(2\pi\sigma^2) - \alpha \frac{1}{2\sigma^2} \sum_{k=1}^{K} (z_{\mathrm{k}} - \bar{z}_{\mathrm{k}})^2 \tag{5.3}$$

$$= -\alpha \frac{K}{2} \ln(2\pi\sigma^2) - \alpha \frac{1}{2\sigma^2} (\boldsymbol{z} - \bar{\boldsymbol{z}})^t (\boldsymbol{z} - \bar{\boldsymbol{z}}), \tag{5.4}$$

which is simply a scaled version of (5.1). Note that the scaling operation changes the gradient and curvature of the log-likelihood, but not the position of the peaks and valleys. Thus, the position estimates, assuming perfect convergence of the solver to a global maximum, are not affected by the scaling operation. In our experiments we found that $\alpha = 0.15$ was a suitable value in all the environments.

5.2 Maximize-Explore

As stated earlier, using GS to obtain a good initial guess on the position estimate can be expensive in terms of computing power. In this section we will develop

[1] In Chapter 6 we will explore two additional methods to solve the over-confidence problem.

two simple heuristics that help a gradient-based solver overcome the local maxima characteristic from our localization problem. These techniques, when used together with a gradient-based method, will offer a less computationally demanding algorithm for solving our localization problem.

As we saw previously, the log-likelihood function acquires large negative values along the robot trajectory, and this can create local maxima in which gradient-based methods might become trapped. Intuitively, the trajectory of the robot acts like a precipice in a hill, which obstructs the path towards the top. Suppose that a gradient-based algorithm becomes trapped in a local maximum created by this precipice: on the one hand the general trend of the likelihood is to increasing in the direction towards the global maximum, while on the other hand we have the precipice stopping the solver. Intuitively, we expect the local maximum to be close to the precipice, and that there must be a point with a higher likelihood *on the other side* of the precipice. This intuition can be grounded again in Panel 5.1b. Our strategy is to exploit this intuition to explore the area close to the local maxima looking for a possible point with a higher log-likelihood than the local maximum, and, if found, to continue running the solver from that location.

The complete algorithm is shown schematically in Table 5.1. It consists of an iteration with two steps: maximization of the log-likelihood and exploration near the solution. The maximization step corresponds to line 4, in which a gradient-based solver is used to find a local maximum starting from an initial position $\vec{\boldsymbol{s}}_0$. The outputs of the solver are a position estimate, $\hat{\vec{\boldsymbol{s}}}$, and the log-likelihood evaluated at that point, l. The exploration step is implemented in lines 5 to 11. In line 5 we generate position samples drawn from the log-likelihood following two alternative techniques that will be explained later. The result is a set of particles $\boldsymbol{S}_t$. In line 6 the function grid_search() evaluates the log-likelihood in all the samples of set $\boldsymbol{S}_t$, and returns the position $\hat{\vec{\boldsymbol{s}}}_c$ with highest log-likelihood l_c. Should this log-likelihood be larger than l, a new iteration begins executing the solver using the new position $\hat{\vec{\boldsymbol{s}}}_c$ as the initial starting point. The iteration will finish when no particle is found during the exploratory phase with a larger log-likelihood than the latest local maximum.

We propose two sample-based search heuristics for the exploratory phase. The idea is to distribute sample points at locations near the local maximum with the following criteria in mind. First, the samples should not be very close to the local maximum in which the solver is trapped. Being the log-likelihood a continuous and smooth function, local maxima are global in a small enough surrounding area, and it is guaranteed that there will not be other local maxima in that area. The problem is, of course, in determining how small this *small enough* area is. But the main idea to retain in the heuristic is that it is highly unlikely to find local maxima very close to each other. Thus, the sampling algorithm should deploy particles with low probability at points close to the local maximum. Second, the

Table 5.1: Maximize-Explore algorithm.

```
 1  maximize_explore(s⃗_0, R, z, θ, n, μ, σ)
 2    ŝ⃗ = s⃗_0
 3    while (true)
 4      [ŝ⃗, l] = maximize_loglik(ŝ⃗, R, z, θ)
 5      S_t = sample_loglik(ŝ⃗, R, z, θ, n, μ, σ)
 6      [ŝ⃗_c, l_c] = grid_search(S_t, R, z, θ, n, μ, σ)
 7      if l_c <= l
 8        break
 9      else
10        ŝ⃗ = ŝ⃗_c
11      endif
12    end
13    return (ŝ⃗)
```

heuristic should deploy samples at points *far enough* as to cross the curve defined by the robot's trajectory (the other side of the precipice). Once more, we have the problem of determining how far is *far enough*.

In order to determine the distances from the local maximum at which to draw samples we will use the information given by the Hessian of the log-likelihood, which is a measure of its curvature. Its inverse is an estimate of the covariance of the position estimate, and its associated ellipse error follows the direction of the contour lines. Thus, this ellipse gives us some useful information about the scale of distances at which to draw particles.

The first proposed exploration heuristic is shown in Table 5.2. It accepts as arguments the current local maximum $\hat{\vec{s}}$, the trajectory of the robot $\boldsymbol{R}$, the RSS observations $\boldsymbol{z}$ and the model parameters $\boldsymbol{\theta}$, all of which are necessary for calculating the log-likelihood and its Hessian. It also requires the number of samples to draw, n, and two extra parameters, μ and σ, that will influence the distance from the local maximum at which the sample particles will be drawn. The algorithm starts by calculating the Hessian of the log-likelihood function in line 2 and the negative of its inverse Q in line 3. Line 4 uses the singular value decomposition of Q to calculate the minor and major semi-axis of its associated ellipse. Instead of using this ellipse directly, we choose an ellipse associated with a confidence interval of 95% probability. Line 5 then calculates ρ, the length of its major semi-axis, by multiplying the largest semi-axis of the ellipse associated with Q by $k_{95} \equiv \sqrt{5.991} = 2.4477$, the multiplying factor necessary to obtain the desired confidence interval. The length of this semi-axis will then be used as a reference distance to draw sample particles in the loop of lines 6 to 10.

The particles are drawn radially from $\hat{\vec{s}}$ with uniform angular probability

Table 5.2: Circular sampling algorithm.

```
circular_sampling(ŝ, R, z, θ, n, μ, σ)
    H = hessian_loglik(ŝ, R, z, θ)
    Q = −H^{−1}
    [U, S, V] = svd(Q)
    ρ = k_95 √max(S)
    for i=1:n
        d = ρ ξ,  ξ ∼ LN(μ, σ)
        θ = 2π rand()
        s_t^i = ŝ + d (cos θ, sin θ)
    endfor
    return ([s_t^1, ..., s_t^n])
```

Table 5.3: Elliptical sampling algorithm

```
elliptical_sampling(ŝ, R, z, θ, n, μ, σ)
    H = hessian_loglik(ŝ, R, z, θ)
    Q = −H^{−1}
    Q = swap_axis(Q)
    L = chol(Q)^t
    for i=1:n
        u = L ζ,  ζ ∼ N(0, I)
        v = k_95 L u/‖u‖ ξ,  ξ ∼ LN(μ, σ)
        s_t^i = ŝ + v
    endfor
    return ([s_t^1, ..., s_t^n])
```

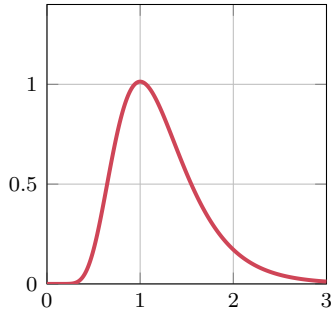


Figure 5.2: PDF of the reference distance correction factor ξ.

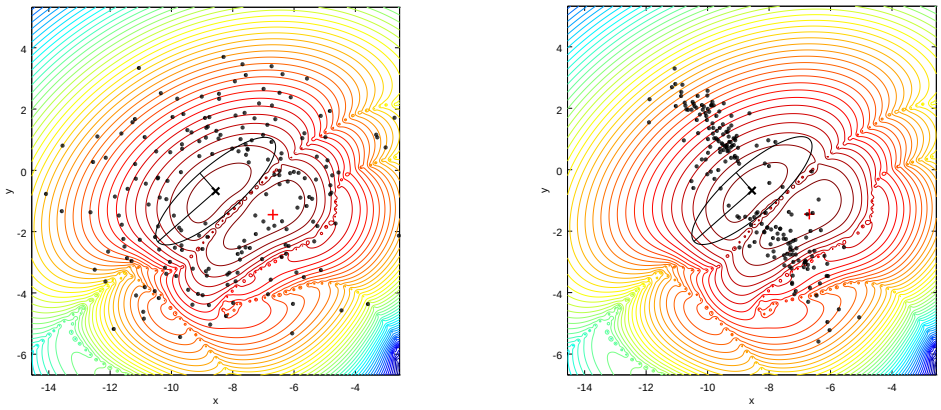
and at distance d, which is calculated as the product of ρ times a multiplicative factor ξ log-normally distributed with parameters μ and σ. The function of the multiplicative factor is to select distances with the criteria explained before: not being *too close* to the local maximum or *too far* form it. This is achieved by selecting appropriate values for μ and σ, which we chose strategically to be $1/e^2$ and $1/e$, respectively. The resulting PDF is displayed in Figure 5.2. The probability of values of ξ being close to zero is small, resulting in a low probability of drawing particles at distances close to zero. The mode is at $\xi = 1$, making the reference distance ρ the most probable. Finally, values smaller than 0.5 and larger than 3 have very small probability, and a large proportion of the probability mass is accumulated between $\xi = 1$ and $\xi = 3$. The result is that the particles will be drawn most probably around the reference distance, not *too close* to the local maximum (low probability of $d < 0.5\rho$) nor *too far* from it (low probability of $d > 3\rho$).

The result of this sampling heuristic when applied to the local maximum of Panel 5.1b is shown in Panel 5.3a. The error ellipse drawn corresponds to a 95% confidence interval, and the dots correspond to the 100 samples generated. Because the samples are drawn with equal probability in any direction, the sampling pattern is circular. We thus refer to this algorithm as a *circular sampler*.

The second sampling heuristic proposed here is a variation of the circular sampler, in which we try to focus the search in directions that are more likely to result in particles crossing the robot trajectory. We achieve this simply by giving more importance in the direction perpendicular to the trajectory of the robot (the precipice). Again, we use the Hessian to obtain information about this perpendicularity.

The proposed algorithm can be seen in Table 5.3. It starts calculating the Hessian H of the log-likelilhood function (line 2) and the negative of its inverse Q (line 3) evaluated in the local maximum. In line 4 we swap the axis of the ellipse associated with Q by exchanging its singular values. This results in an ellipse perpendicular to the original one, whose major axis will hopefully be perpendicular to the robot trajectory. The idea now is to use this ellipse to generate samples principally in the direction of its major axis.

The n samples are drawn in the loop of lines 6 to 10. In line 7 we generate a sample from a bivariate normal distribution with covariance Q. We do it by multiplying L by an uncorrelated sample $\vec{\zeta} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$, where L is the Cholesky factor of Q pre-calculated in line 5 for convenience. In line 8 we select the distance of the particle to $\hat{\vec{s}}$ multiplying L by k_{95} and ξ, again a log-normally distributed random variable with the same μ and σ as in the circular sampler. The matrix L plays the equivalent role as ρ in the circular sampler, this time providing the reference lengths in two dimensions instead of only one. The vector $\vec{u}$ is divided by its module, and therefore it does not affect the distance. The important



(a) Circular sampler.

(b) Elliptical sampler.

Figure 5.3: Samples drawn using the exploration heuristics.

information that $\vec{u}$ carries is the direction in which to draw the particle, which favours those more parallel to the major axis of Q.

The result of applying this particular sampling heuristic to the local maximum of Panel 5.1b can be seen in Panel 5.3b. As we can see, the samples are placed within an elliptic area while leaving the central part rather empty. We thus refer to this algorithm as an *elliptical sampler*.

Either the circular or elliptical samplers can be used with the Maximize-Explore algorithm. Which one to use then? Panels 5.3a and 5.3b show the results of the samplers in a sample case in which there is a clear dominant direction of the covariance error ellipse. In this exemplary case both samplers allocate approximately half of the particles on each side of the robot trajectory, and thus both perform equally well in the sense that the probability of allocating particles on the other side of the robot trajectory is rather similar. In some other practical cases the resulting covariance ellipse will not be as well aligned with the robot trajectory as in our sample case. Depending on the case, it might be better to use one or the other sampler. If the trajectory of the robot does not have many intersections or points close to previously visited areas, then the elliptical sampler might require less samples to achieve the same probability of placing particles cross the precipice. The circular sampler, however, uses less geometrical information than the elliptical one, and thus it is more robust against wrong assessments related to the existence or not of favourable conditions towards using the elliptical one. All of these assessments are done based on common sense grounded in experimental tests with our real data.

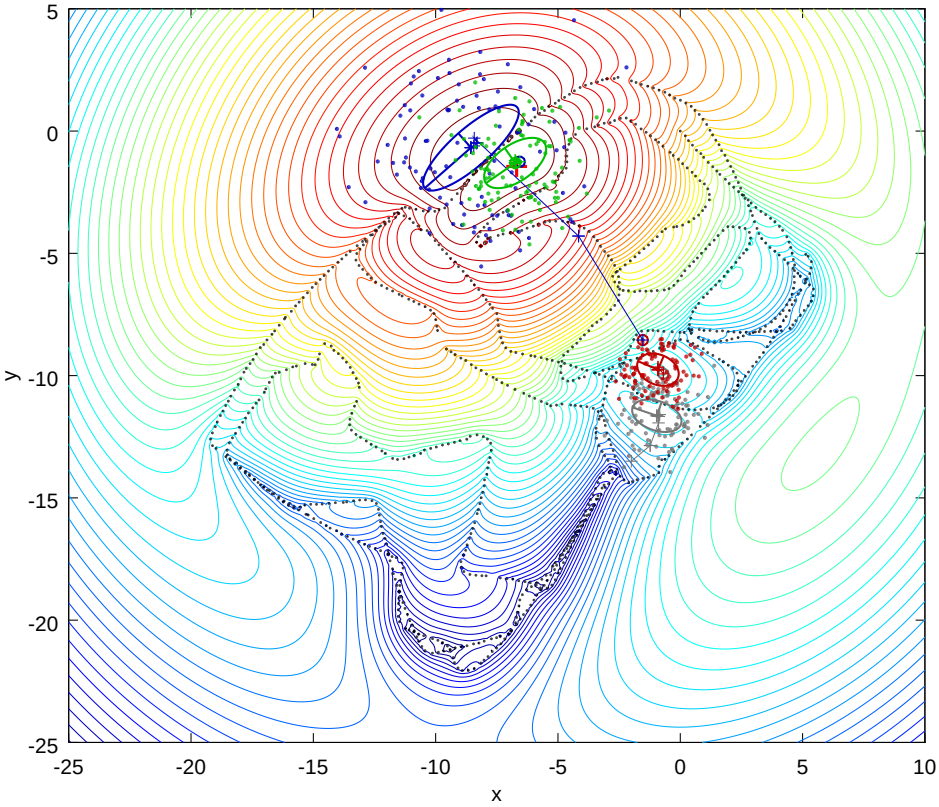


Figure 5.4: Maximize-Explore example.

Figure 5.4 presents an illustration of how the Maximize-Explore algorithm behaves in practice. In the example, we used a L-BFGS-B solver in the maximization step and a circular sampler drawing 100 particles in the exploratory phase. The evolution of the algorithm is represented with a different colour for each iteration. Thus, we can distinguish four iterations. The algorithm was started in the position $(-2, -13.5)$, and the data associated with the first iteration is marked in grey colour. In each iteration, the evolution of the solver corresponding to the maximization step is marked with a solid thin line; the intermediate steps of the solver are marked with a + sign on top of it. Upon convergence, a 95% confidence error ellipse is plotted centred within the local maxima found by the solver. The particles drawn by the sampler are marked with dots, and the selected sample chosen to start the subsequent iteration, if any, is marked by surrounding it with a small circle.

As we can see, in this case the algorithm was capable of overcoming the three local maxima in which the solver become trapped. The global maximum was reached in the fourth iteration, marked in green. In the next to the last iteration (marked in blue), the result of the exploratory phase is a particle very close to the global maximum, and thus the evolution of the solver in the last iteration (in green) is not perceptible. We also can see that the exploratory phase corresponding to this last iteration did not find any continuation candidate (there is not a green particle surrounded by a green circle), which caused the algorithm to stop.

It is worth noting that Figure 5.4 presents one realization of the algorithm, and that, given the random nature of its exploratory heuristic, different realizations could result in different evolutions becoming trapped in different local maxima. The example presented here is an illustration of a visually clear case. Also, the Maximize-Explore algorithm does not guarantee convergence to a global maxima, but, as we have seen, it simply helps the solver overcome the type of local maxima that we have typically found in our localization experiments. The larger the number of samples used during the exploration phase, the more certain we will be about the ability of the algorithm to overcome local maxima, but at the price of more computing power being used. Should we increase the number of samples significantly, at some point the number of log-likelihood evaluations can grow and be equivalent to using a GS-based method. The advantage that the Maximize-Explore method still holds in this case is that the GS-based method needs the definition of a grid, whereas the Maximize-Explore algorithm can be started in a (reasonable) random point. This point could be calculated using e.g projection onto convex sets [Hero and Blatt, 2005, Gholami et al., 2011] or the linear position estimators proposed in Section 4.5.

Finally, in order to obtain a practical performance comparison between the L-BFGS-B initialized with a grid search (GS-LBFGSB) and Maximize-Explore algorithms, we again used the sample case presented in Figure 5.4 as a reference problem. The grid used in the first algorithm covered reasonably well the area containing all the nodes (30x25m^2) with a resolution of 0.5 m. In total the grid had 3111 points, resulting in a computational cost of 3111 evaluations of the log-likelihood function plus one execution of the L-BFGS-B solver. To evaluate the performance of Maximize-Explore, we ran the algorithm ten times, all starting from the same initial position as in Figure 5.4 and using a circular sampler drawing 100 particles. It converged to the global maximum eight out of the ten trials, and on average it was 7.6 times faster than the GS-LBFGSB. This comparison was calculated using the CPU time required by both algorithms as reported by Octave. The algorithms, while being different, were implemented using the same core subroutines, and thus the comparisons, while not being strictly accurate, are valid for a first approximation. Taking as a concrete example the realiza-

tion shown in Figure 5.4, we can still argue that it required four iterations to converge, and thus 400 evaluations of the log-likelihood function were necessary, plus four executions of the L-BFGS-B solver. Thus, the ratio 3111/400=7.77 approximately gives the 7.6 times increase in performance of the Maximize-Explore algorithm, with a small margin due to the extra L-BFGS-B executions. In any case, this ratio confirms a significant computational saving of Maximize-Explore over a GS-based method.

5.3 Localization Using Common and Individual Models

In Chapter 3 we described how to identify the log-normal model and studied the variability of its parameters depending on the nodes used for the identification. In view of this variability, we argued that using IMs results in better overall RSS predictions with respect to CMs, and we hypothesized that this would lead to better position estimates. In this section we present experimental results comparing the usage of individual versus common models for localization using the ML and LLS position estimators studied in Chapter 4.

5.3.1 Performance Criteria

In order to compare the different types of models and localization methods, it is necessary to first define a performance measure that we can use to make the comparisons. We choose as a metric the average localization error resulting from using part of our experimental data for model identification and the rest for localization. This eliminates the possibility of having unrealistically small localization errors due to model over-fitting. We now specify how we calculate this error in practice.

Let us start by defining a measure of errors associated with using CMs identified with data from a particular set of nodes. Let us call $\mathcal{N} = \{n_1, \ldots, n_M\}$ the set of all the M available nodes, and $\mathcal{S}$ a subset of $\mathcal{N}$ containing L selected nodes ($L \leq M$). We identify one CM per robot trajectory using the data coming from the selected nodes. Thus, we have three independently identified models. For each of those models we still have data from the other two robot trajectories that have not been used for the model identification. We then use those datasets to estimate the position of all the nodes. Thus, we have $2 \cdot M$ position estimates per model, and in total $3 \cdot 2 \cdot M$ for the three CMs. The localization error associated with the set $\mathcal{S}$ is then the average distance between the true and all the estimated node positions. Referring to $k^{\complement} = \{1, 2, 3\} \setminus k$ as the set of integers from one to

three excluding k (so that $k \cup k^{\complement} = \{1, 2, 3\}$), we can write this error as:

$$e_{\mathcal{S}} = \frac{1}{3 \cdot 2 \cdot M} \sum_{k=1}^{3} \left[\sum_{j \in k^{\complement}} \left(\sum_{i=1}^{M} \left\| \vec{\boldsymbol{s}}_{\mathrm{i}} - \hat{\vec{\boldsymbol{s}}}^{\mathcal{S}}_{\mathrm{ijk}} \right\| \right) \right], \tag{5.5}$$

where $\vec{\boldsymbol{s}}_{\mathrm{i}}$ is the true i-th node position and $\hat{\vec{\boldsymbol{s}}}^{\mathcal{S}}_{\mathrm{ijk}}$ its estimate using the data from the j-th robot trajectory for localization and a model identified using the data from the set of nodes $\mathcal{S}$ collected during the k-th robot trajectory.

To compare the common and individual models, we choose CMs identified with the data collected from all the nodes ($\mathcal{S} = \mathcal{N}$). Recall that we referred to these models as reference models in Section 3.2. Reference models contain all the information from the training data, and therefore they can be thought of as being the best CMs in terms of explaining the training data. For future reference, we denote the error calculated using reference models and equation (5.5) as **reference errors**. Note that, although we have three reference models, one per robot trajectory, using the reference models to calculate the error using (5.5) results in a unique reference error for each localization algorithm used in each of our experimental environments.

Regarding the calculation of errors associated with IMs, the formula is the same as for CMs, but this time using IMs for localization. In other words: instead of using the same model to locate all the nodes, we will use for each node its own associated model. Therefore, the term $\hat{\vec{\boldsymbol{s}}}^{\mathcal{S}}_{\mathrm{ijk}}$ from (5.5) now represents the position estimate of node i using the data from the j-th robot trajectory for localization and a model identified using the data from the i-th node (itself) collected during the k-th robot trajectory (that is, $\mathcal{S} = \{n_{\mathrm{i}}\}$).

5.3.2 ML Localization Results

Table 5.4a shows the results of using ML localization for both individual and common models. In all cases the solver used was the BFGS with a grid search initializer with grid a resolution of 0.5 m. The errors reported are calculated using the cross-validation scheme explained in Subsection 5.3.1.

As expected, the smallest errors are attained in the basketball field, where there are less obstructions and the measurement model is less biased. The largest errors occur in the lobby, despite being a rather open space compared to the office. We believe that the reason for this result is that the environment is less homogeneous, with some large dominant obstacles (see Section 1.4). As discussed in Section 2.6, radial models become worse predictors when the environment is more inhomogeneous. It is also interesting to point out, however, that, from (4.45), we can see that the CRLB of position estimators using the log-normal model is proportional to $(\sigma/n)^2$ through γ (see its definition in (4.9)). In the case

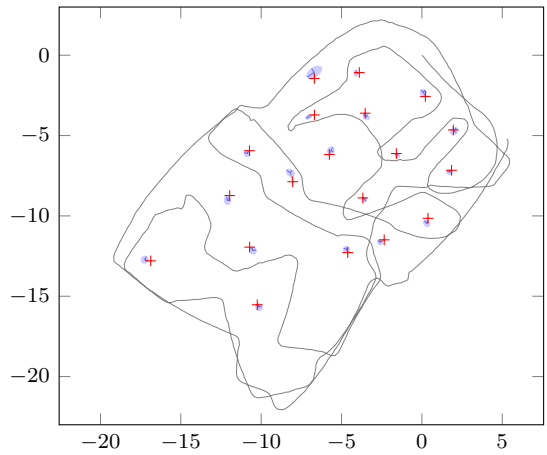
Table 5.4: Comparison of average errors of ML and LLS position estimates in different environments. CM=common model, IM=individual model. Distance in metres

(a) ML

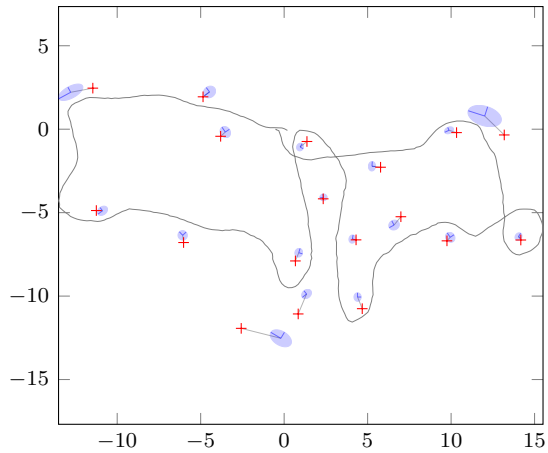
	CM	IM	Ratio(CM/IM)
Basketball field	0.94	0.33	2.83
Lobby	1.44	1.07	1.35
Office	0.97	0.78	1.25

(b) LLS

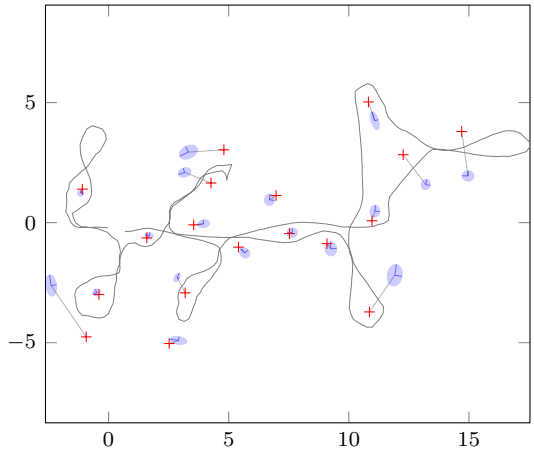
		CM	IM	Ratio(CM/IM)
OLS-ML	**Basketball field**	4.43	3.05	1.45
	Lobby	21.10	56.24	0.38
	Office	5.44	4.91	1.11
OLS-Unbiased	**Basketball field**	3.25	1.72	1.89
	Lobby	13.33	23.98	0.56
	Office	3.18	2.54	1.24
OLS-LMMSE	**Basketball field**	3.79	3.05	1.24
	Lobby	7.01	8.85	0.79
	Office	3.07	2.72	1.13
GLS-ML	**Basketball field**	0.67	0.71	0.94
	Lobby	2.07	2.14	0.97
	Office	1.32	1.31	1.01
GLS-Unbiased	**Basketball field**	0.87	0.86	1.01
	Lobby	2.56	2.56	1.00
	Office	1.50	1.44	1.04
GLS-LMMSE	**Basketball field**	1.13	1.06	1.06
	Lobby	3.19	3.07	1.04
	Office	1.73	1.63	1.06
IGLS-ML	**Basketball field**	0.46	0.38	1.22
	Lobby	1.27	0.98	1.30
	Office	0.78	0.65	1.19
IGLS-Unbiased	**Basketball field**	0.55	0.50	1.11
	Lobby	1.43	1.37	1.04
	Office	0.90	0.77	1.16
IGLS-LMMSE	**Basketball field**	1.14	0.85	1.34
	Lobby	2.58	2.30	1.12
	Office	1.43	1.31	1.09



(a) Basketball field.

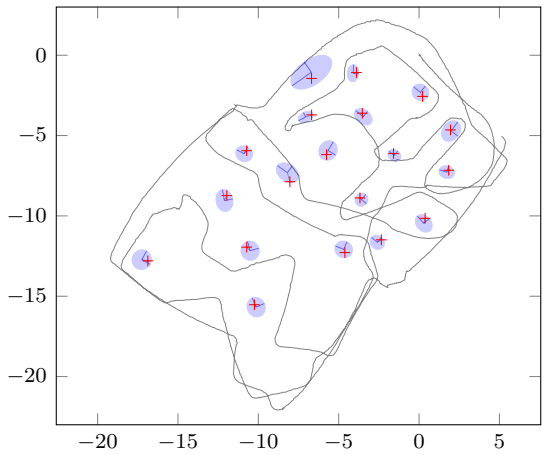


(b) Lobby

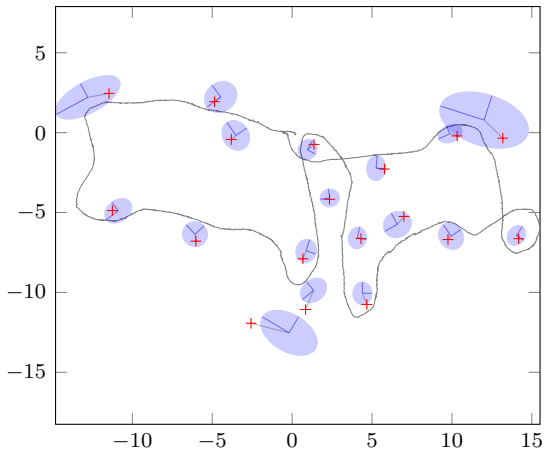


(c) Office

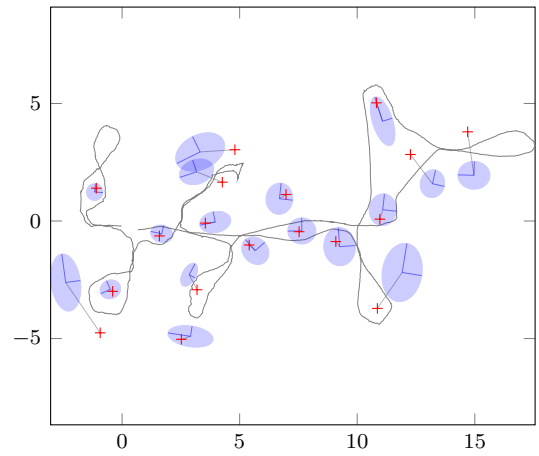
Figure 5.5: Example of ML localization results (no overconfidence compensation).



(a) Basketball field.



(b) Lobby



(c) Office

Figure 5.6: Example of ML localization results with overconfidence compensation.

of our experiments, the average values of σ/n for the common reference models are 1.77, 1.90 and 1.86 for the basketball field, lobby and office respectively, suggesting that good unbiased position estimators in the lobby are naturally less accurate.

It is especially relevant to note the difference between the errors when using the reference common model and individual models. Clearly, using IMs results in smaller average errors in all the environments with respect to using CMs. This is a direct consequence of the variability of the model parameters observed in Section 3.2. However, the relative improvement is not the same in all environments. Table 5.4a also contains a ratio of the errors for a better dimensionless comparison of the enhancement. This ratio shows an error reduction of up to 65% when using IMs compared to using CMs in the basketball field, which is significantly larger than the reductions for the lobby and the office (26% and 20% respectively). A more detailed discussion regarding the benefits of using IMs will be presented in Subsection 5.5.2. In any case, these results motivate us to study in more detail the sensitivity of ML position estimates with respect to variations of the model parameters, which will be explained in Section 5.4.

Figures 5.5 and 5.6 show an example of the localization results obtained using IMs and data collected from one of the robot trajectories for each environment. Figure 5.5 presents the results when using plain ML, whereas Figure 5.6 when using the over-confidence compensation method described in Section 5.1. The solid thin black curves represent the robot trajectory used to collect the data for localization in the particular examples displayed. The true nodes' position is marked with a red '+' sign. The error ellipses are centered in the estimated positions, and are drawn for a 95% confidence interval. Clearly, without over-confidence compensation the true positions are outside the error ellipses in most cases.

5.3.3 LLS Localization Results

In this subsection we assess the performance of the three different families of RSS-based linear position estimators studied in Chapter 4 using our experimental data. In contrast with ML, these all have the great advantage of not requiring an initial position to start the algorithm. Furthermore, some of them are closed form formulas that do not require iterations, being a very fast alternative to ML. On the other hand, they produced less accurate results. In that case, if a finer accuracy is required one can always use them as a deterministic method to calculate the initial position with which to start a ML solver. All in all, they are promising algorithms that we want to validate with real data.

Recall that we have three families of linear algorithms according to the solver used: OLS, GLS and IGLS. In each family, depending on the distance estimator

Table 5.5: Modified IGLS algorithm.

```
 1  iglse(R, d̂_m, σ)
 2    A_L = f(R)
 3    b̂_L = f(d̂_m, R)
 4    d̂_mc = d̂_m
 5    d̂_0 = (d̂_m)^{1/2}
 6    repeat
 7      Σ̂_L = f(d̂_mc, σ)
 8      ŝ = [A_L^t Σ̂_L^{-1} A_L]^{-1} A_L^t Σ̂_L^{-1} b̂_L
 9      d̂ = (||ŝ − r_1||, ..., ||ŝ − r_K||)^t
10      d̂_mc = (d̂ + α|d̂ − d̂_0|^2)^2
11    until convergence
12    Σ̂_L = f(d̂_m, σ)
13    Cov[ŝ] = [A_L^t Σ̂_L^{-1} A_L]^{-1}
14    return (ŝ, Cov[ŝ])
```

used, we have three additional possibilities: ML, unbiased and LMMSE. Therefore, we have in total nine possible linear position estimators. Table 5.4b shows the results of calculating the localization error using common and individual models with all of them.

Among the OLS methods, once more the one using LMMSE distance estimators is the best option. Note, however, the generalized large increase in accuracy of the GLS compared to the OLS methods. Once more, as predicted in the simulations, the GLS-ML is the best one among the GLS methods.

The IGLS algorithm implemented as in Table 4.2 in general performed significantly worse than expected based on the simulations, and clearly worse than the GLS methods. The reason for this was that in some punctual cases there were very large errors in the order of hundreds of metres which greatly affected the average error. For some reason, the position estimates were sometimes drifting far from the position estimated by the GLS algorithms. The problem clearly had to do with the iteration of the IGLS method.

Note that after the first iteration the calculation of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ does not depend directly on the original node-beacon distances estimated from the RSS measurements, but on the distances between the last position estimate and the beacons (see Table 4.2). Thus, the calculation of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ is linked to the observations only indirectly through the position estimate that uses $\hat{\boldsymbol{b}}_{\mathrm{L}}$ in line 6. This *decoupling* seems to be the cause of the occasional large errors encountered.

In order to solve this issue we modified the original algorithm to the one shown in Table 5.5. The main difference is the modification of the distances

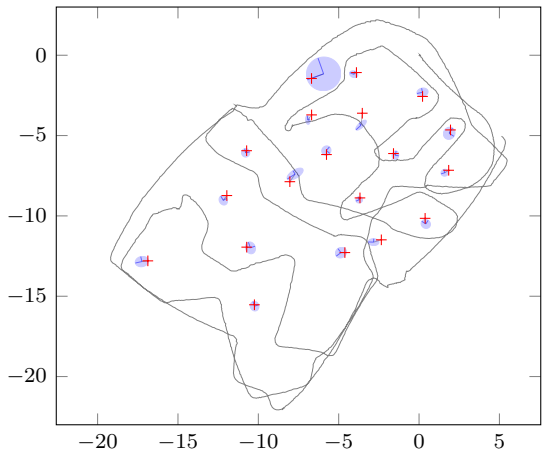
used to calculate $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$. The modified algorithm uses a corrected vector of squared distances $\widehat{\boldsymbol{d}}_{mc}$ which is calculated adding a correcting factor $\alpha|\hat{\boldsymbol{d}}-\hat{\boldsymbol{d}}_0|^2$ to the latest vector of distance estimates $\hat{\boldsymbol{d}}$ (see line 10 of Table 5.5), where α is a positive scalar and $|\hat{\boldsymbol{d}}-\hat{\boldsymbol{d}}_0|$ represents the element-wise absolute value of the vector $\hat{\boldsymbol{d}}-\hat{\boldsymbol{d}}_0$. The vector of distances $\hat{\boldsymbol{d}}_0$ is defined in line 5, and corresponds the original vector of distance estimates calculated from the input vector of squared distance estimates $\hat{\boldsymbol{d}}_m$. The exponentiation in lines 5 and 10 are element-wise. The correcting factor is then a vector formed by non-negative elements which grow with the difference between their respective initial and latest distance estimates. The vector $\widehat{\boldsymbol{d}}_{mc}$ will be used in the computation of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ in the next iteration in line 7, coupling directly the calculation of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ with the original distance measurements $\hat{\boldsymbol{d}}_0$.

We now explain the rationale behind the proposed corrective technique. From the formulas of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ presented in Section 4.5, we can see that $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ depends on the 4-th power of the distances used to compute it, and therefore the correction factor increases the elements of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$. The inverse of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ is the weighing matrix used to solve the GLS problem in line 7. Consequently, an increase of the i-th element of the vector $\widehat{\boldsymbol{d}}_{mc}$ will imply a reduction of its associated weight, and therefore a decrease of its relative importance in the estimator $\hat{\vec{\boldsymbol{s}}}$ with respect to the rest of the distance estimates. Therefore, the correction factor penalizes position estimates whose node-beacon distances are far from the original estimated distances $\hat{\boldsymbol{d}}_0$.

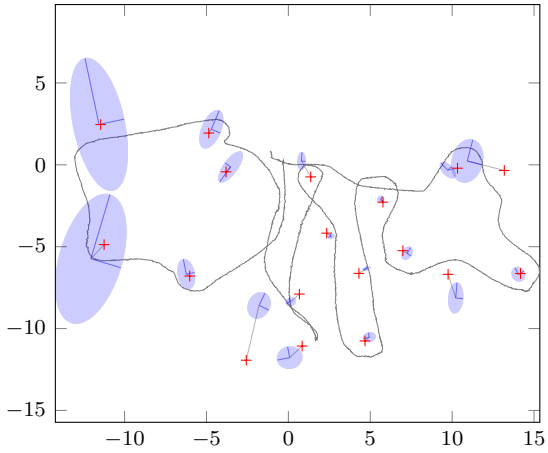
The selection of a quadratic function for the correction factor is a natural and common choice when one wants to penalize deviations with respect to a reference value. We have also tried different polynomials of the type $\alpha|\hat{\boldsymbol{d}}-\hat{\boldsymbol{d}}_0|^r$ with several values for r, including the linear case ($r = 1$). Note that the penalty increases quickly with the exponent r, and the elements of $\hat{\mathbf{\Sigma}}_{\mathbf{L}}$ even faster as they depend on the 4-th power of $\widehat{\boldsymbol{d}}_{mc}$. This might result in non-positive definite matrices when calculating $\hat{\mathbf{\Sigma}}_{\mathbf{L}}^{-1}$ due to round-off errors. We have found experimentally that a squared function is a good compromise, with values for α between 0 and 1. The results for the IGLS algorithm shown in Table 5.4b were calculated using $\alpha = 0.2$. This value was found to be the best one in the sense that it minimized the average CM and IM errors as defined in (5.5).[2] Examples of localization results using the IGLS using ML distance estimators (IGLS-ML) algorithm in the three environments can be seen in Figure 5.7.

The localization results when using IGLS estimators are interesting. Remarkably, the average localization errors are comparable with using ML, and even smaller in most cases when using ML distance estimators (see Table 5.4a and 5.4b). Based on the simulations of Chapter 4, we were expecting the ML to be the best method in general. It is the case when using IMs in the basketball field. But surprisingly the modified IGLS-ML method performs very similar to the ML in all the environments when using IMs, and clearly better when using CMs.

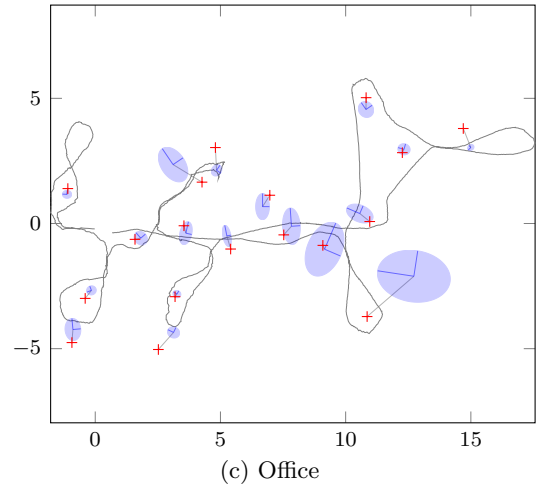
[2]We still encourage the reader to try different values in their own settings.



(a) Basketball field.

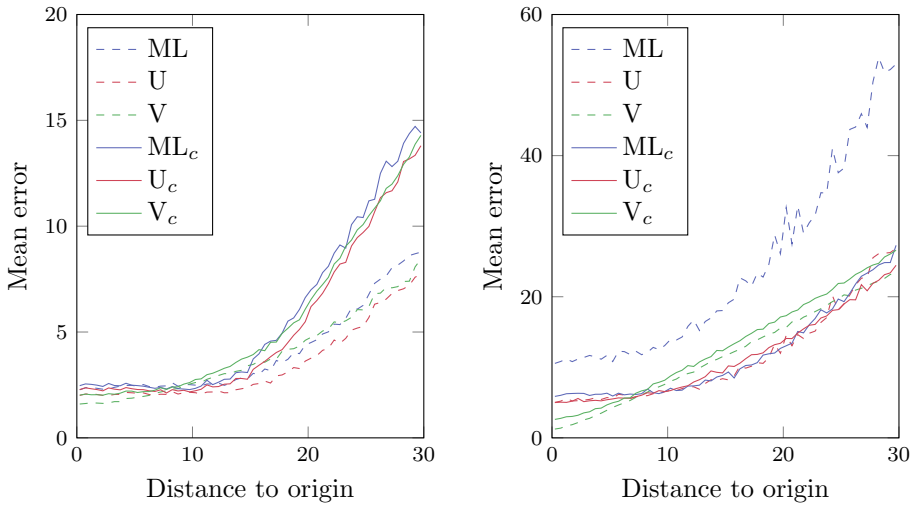


(b) Lobby



(c) Office

Figure 5.7: Example of IGLS-ML localization results.



(a) $\sigma = 2$ (b) $\sigma = 5$

Figure 5.8: Comparison between the original and modified IGLS methods. The c subindex denotes the corrected algorithm. Model parameters: $z_0 = -70$, $n = 2$, $\sigma = 2$ (left) and $\sigma = 5$ (right).

When comparing among the IGLS algorithms, clearly IGLS-ML is the best (see Table 5.4b). This is not what the simulations of section 4.7 predicted when the original IGLS algorithm was used. To study this difference we run the same simulations again comparing both versions of the IGLS algorithms. The results are shown in Figure 5.8. For low noise levels, the corrected versions have equal or slightly worse mean error close to the origin of coordinates (see Panel 5.8a). After the breakpoint of d=15 m, the corrected versions clearly perform worse. However, for larger noise levels the corrected methods perform rather similarly as the original ones when using the unbiased and LMMSE distance estimators, but there is a remarkable improvement in the ML method (see Panel 5.8b).

Although the IGLS-ML algorithm performs the best in terms of the smallest average error, it is worth considering the advantages of the GLS-ML method. As with the IGLS, it does not need an initial guess at the position estimate. However, its solution is not iterative and requires only solving a linear system. Yet it had a localization error of approximately 70 cm in the basketball field, 2 m in the inhomogeneous lobby and 1.30 m in the office. This mixture of low computing power and relatively good accuracy makes it a good candidate for applications with constraints in computational resources or power consumption.

5.4 Sensitivity of ML localization to Variations of the Model Parameters

When using a site-specific RSS-distance model for localization, the performance of the algorithm depends highly on the validity and appropriateness of the model. Deviations in the model parameters with respect to the true value provoke in turn deviations in the position estimates. Therefore, the variability of the model parameters has a direct impact on the performance of position estimators.

As we have seen in section 3.2, the values of the parameters can vary significantly depending on the nodes used to identify the models. A natural question that arises then is how much the performance of a localization algorithm is affected as a consequence of this variability.

Studying the sensitivity of localization algorithms on the model parameters is not straightforward, because it depends on the measurements model, the geometrical configuration of the problem and on the algorithm itself. As a first approximation, the sensitivity associated with using a specific model could be studied by investigating how the CRLB is affected by changes in the parameters. For example, in [Salman et al., 2012] the authors analyze via simulations how the CRLB is affected by deviations of the path loss exponent with respect to the true value when the reference power is known. However, as we have seen, the RSS-distance model is in general biased, and so are the position estimators that we have studied. Because of the many factors that affect the sensitivity and the impossibility to calculate the bias, we opt for an empirical approach.

In this section we present a quantitative empirical study on the sensitivity of ML localization with respect to variations in the parameters of a log-normal common model. The main idea is to compare the localization error when using a well-identified CM with that when using other CMs whose parameters are changed artificially. Our choice for this well-identified model is the reference models from Section 3.2, which recall that are identified using the data collected from all the nodes in a robot trajectory. Recall also that we denoted the error associated with using the reference models in each environment as *reference errors* (see Subsection 5.3.1).

In our study we proceed as follows. First we modify the parameters of the reference models multiplying them independently by factors ranging from 0.5 to 2. Then we calculate the new localization error associated with the modified models using (5.5), and we then compare it with the reference error. Figure 5.9 shows a contour plot of the sensitivity of the ML position estimates in the different environments. The horizontal and vertical axes show the multiplication factors k_n and k_{z0} used to modify n and z_0, respectively. The area covered in Figure 5.9 corresponds to a variation of the parameters similar to the one experienced in practice, which can be appreciated in figures 3.2 and 3.3. The

point corresponding to the unmodified reference models is the one with $k_n = 1$ and $k_{z0} = 1$, which we will refer to as the **reference point**. The resolutions for variations of k_n and k_{z0} are 0.03 and 0.01 respectively. The colour bars present the localization errors normalized with their corresponding reference error.

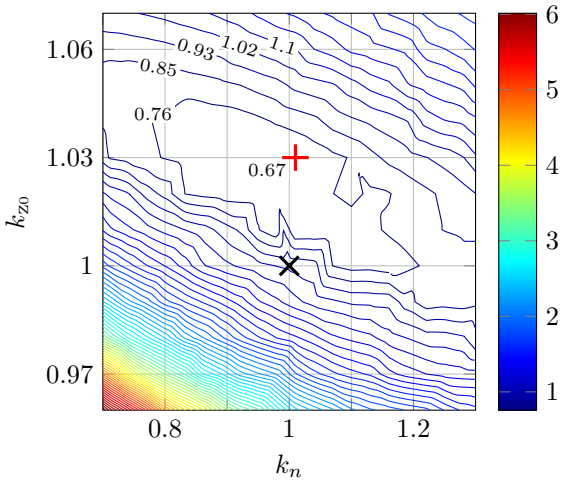
Figure 5.9 should be then interpreted as follows: the horizontal and vertical axes represent the normalized percentages of changes in magnitude of the model parameter values, and the contour lines the relative change of the localization error with respect to the reference error as a result of those changes. For example, if the magnitudes of n and z_0 are increased by 20% ($k_n = 1.20$) and 3% ($k_{z0} = 1.03$) respectively, the localization error in the basketball field is approximately 93% of the reference error.

Variations in σ do not impact the result. This can be seen straight in the equations of the log-likelihood, for example in (5.3). As we can see, σ affects the log-likelihood function through multiplicative and additive factors that scale and add a constant to the log-likelihood, respectively. These operations do not change the location of the maxima, and assuming perfect convergence of the solver to the global maxima, do not affect the position estimates. This is, however, a direct consequence of the IID measurements assumption. In the more general case of having a heteroskedastic process, the measurement's covariance matrix, although diagonal, will not contain equal terms, and thus we cannot factor out σ as we do when assuming IID measurements. In practice, this case arises when estimating the node's position using individual models for the beacons,[3] case in which having IMs with different values of σ will indeed affect the estimated position.

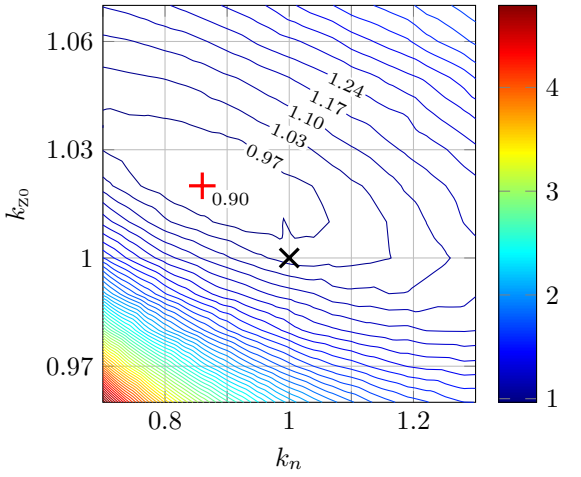
Figure 5.9 shows that the normalized errors can change significantly when modifying the model parameters. Relative variations of z_0 in general have a notably larger impact than relative variations of n. The curvature of the error is higher in the basketball field and lobby, suggesting that the relative sensitivity is higher in open environments. Interestingly, we can see that sometimes artificial variations of the model parameters can be beneficial. The minima (red plus signs) are not attained with the unmodified models (black crosses). We obtain them with parameter variations corresponding to $k_n = 1.01$ and $k_{z0} = 1.03$ for the basketball field, $k_n = 0.86$ and $k_{z0} = 1.02$ for the lobby and $k_n = 0.74$ and $k_{z0} = 1.00$ for the office. These result in error reductions of 33%, 10% and 21% respectively (average errors of 0.63 m, 1.29 m and 0.76 m respectively).

The impact of the parameter variations depends on the direction of change. When considering variations near the reference point (black crosses), we can see that relative reductions of the parameters (in magnitude) can increase the error rapidly, especially in the basketball field and lobby. In particular, relative variations of z_0 can have a strong effect on the error. As we discussed in section

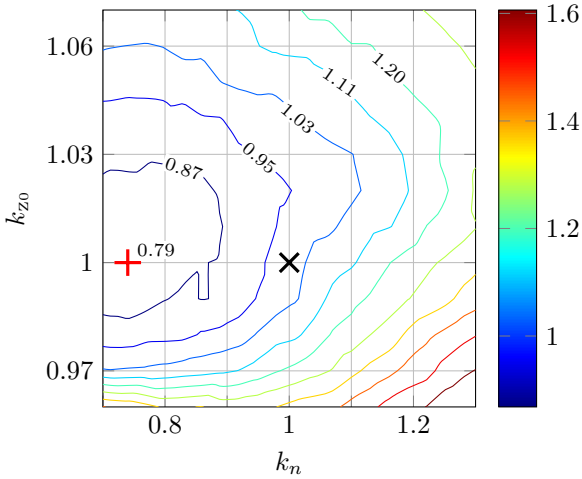
[3]Note that so far in this thesis we have considered using individual models for the nodes whose position is to be estimated, and not for the beacons.



(a) Basketball field.



(b) Lobby.



(c) Office.

Figure 5.9: Relative sensitivity contour plots.

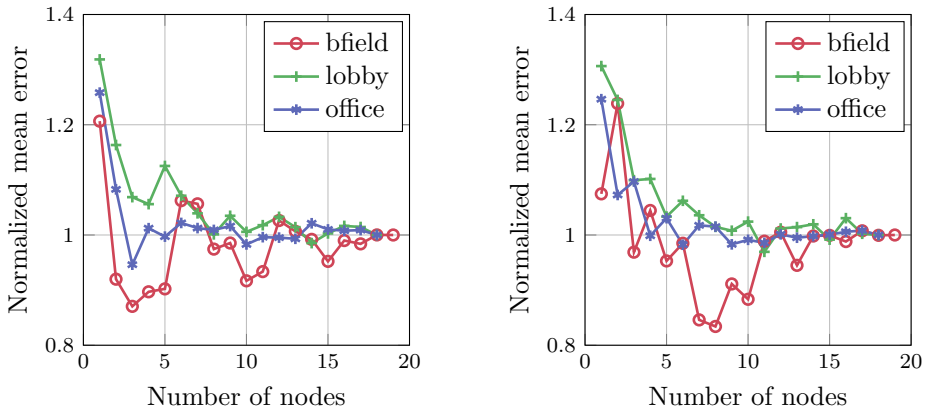


Figure 5.10: Mean error of ML position estimates when using a common model depending on the number of nodes used in the model identification L. The figure presents two executions of the algorithm.

3.2, among the three environments the basketball field is the one where the impact of the environment on the variability of the parameters is the smallest. The other lecture of this is that the HW variability has the largest relative impact in open environments. Because the lobby and office contain more obstacles, the relative contribution of the HW variability to the model uncertainty diminishes compared to the contribution of the environmental inhomogeneity. This, together with the larger normalized sensitivity in the basketball field shown in Panel 5.9a, suggests that the HW variability can have a relatively important impact on the localization performance in open spaces.

5.5 Model Identification Using Common and Individual Models

In this section we study and elaborate on how the model identification using common and individual models affects the performance of ML localization.

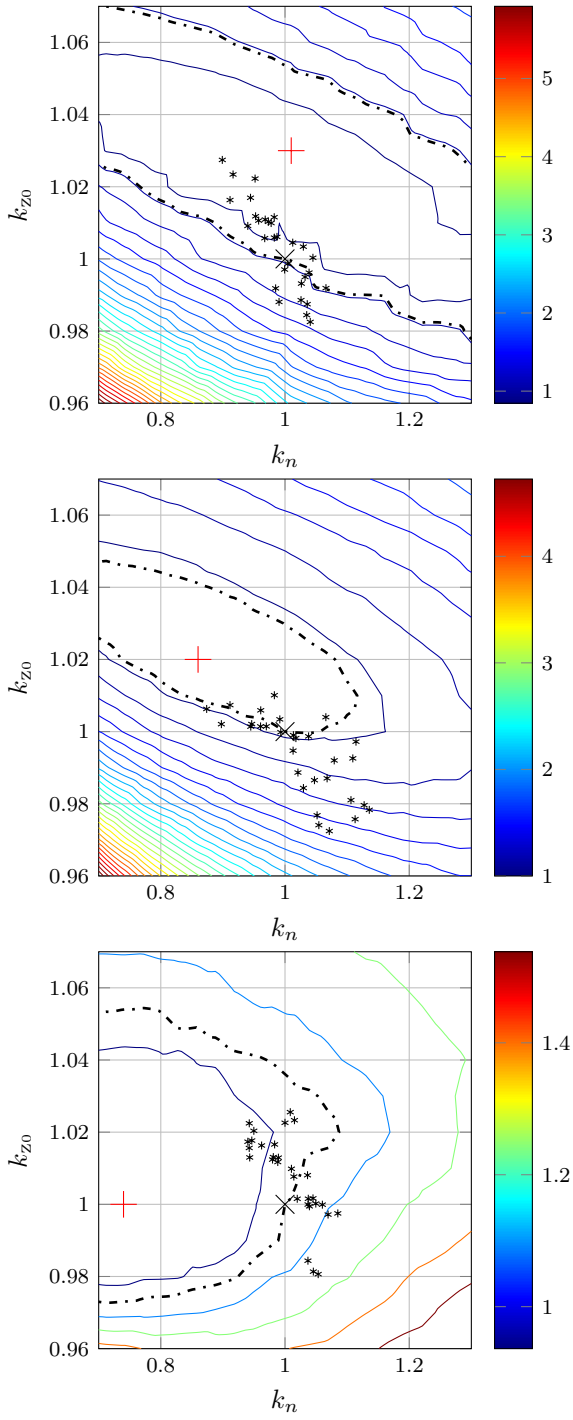
5.5.1 Model Identification Using a Common Model

Next, we study how using a different number of nodes during the model identification affects the localization error as a consequence of variations in the model parameters. In order to make this study we identify CMs using data from different sets of nodes, for which we calculate their associated error using (5.5).

The sets contain a number of nodes between one and all the available units ($L = \{1, .., M\}$). For each number of nodes used in the model identification, L, we choose (up to) ten different combinations of L randomly selected nodes. For each of these sets we calculate its corresponding localization error. Finally, we calculate the average of all the errors associated with the ten sets and normalize it with the corresponding reference error for each environment. We then plot the normalized error as a function of the number of nodes used in the model identification, L. Due to the fact that the combination of nodes are randomly selected, different runs of this procedure will yield different results. Figure 5.10 shows the results of two independent executions.

Although the curves differ for different executions of the procedure, we can see a clear tendency: an initial relative error reduction and a stabilization as the number of nodes used in the model identification increases. When using fewer nodes, the model parameters can reach areas *far* from the reference point (see Fig.5.9). If, by chance, we select the right set of nodes, the model might reach points close to the minimum. But, without *a priori* information, we have a high risk of selecting nodes whose models fall in regions with a high normalized error. Using more nodes to identify the model, on the other hand, *attracts* the parameters towards the reference point (black crosses), making them fall closer to it and diminishing the risk of reaching areas with large errors. Therefore, using more nodes to identify CMs results in more robust models against errors resulting from HW variability and local environmental inhomogeneity.

This can be better explained in view of Figure 5.9. Suppose we use data from only one node, say node a, to identify a single model M_a. This model can have its representation at any point of the figure, and therefore we can expect a large variability of its associated error depending on the area where the model falls into. It might well hit the point with the smallest error, but given the shape of the error function, there is a high risk that it will end up having a large error. Suppose that we select another node, say node b, to identify a different model, M_b. Again, this model can have its representation in virtually any point in Figure 5.9. Suppose now that we identify one common model M_c using the data from nodes a and b. Intuitively, its parameters will be between the parameters of M_a and M_b. Thus, the area of Figure 5.9 in which M_c has its representation becomes more restricted. In this way, we expect that the area in which the model parameters can fall becomes smaller as the number of nodes used for the model identification increases. Eventually, when the data from all the nodes are used for the model identification, the area will converge to the reference point. The decrease in the size of the area where the model has its representation results in lower variability of the error, which tends to become closer to the reference error as L increases. And this lower variability decreases the risk of the model falling into areas with large errors.



(a) Basketball field.

(b) Lobby.

(c) Office.

Figure 5.11: Model parameters for the common-models identified using 5 nodes corresponding to Figure 5.10 on the left.

To illustrate more clearly the reason behind the variability observed in Figure 5.10, we have prepared Figure 5.11, in which we reproduce graphically the situation that lead to the results seen at the left of Figure 5.10 when using $L = 5$. As we can see, the normalized errors are significantly distinct for the different environments. While the error for the office is similar to the reference error, the errors for the basketball field and lobby are smaller and larger, respectively. The small black asterisks in Figure 5.11 represent the selected models with which the average errors were calculated. The dotted line represents the contour line with a normalized error equal to one (with respect to the reference error). Thus, points within the area enclosed by that contour line contribute to reducing the average error with respect to the reference error, whereas points outside increase it.

In the basketball field the majority of the models fall within the area inside the contour ellipse, and thus the resulting relative error is lower than 1. In the lobby, on the other hand, most of the dots are outside the contour line, contributing to a relative error greater than 1. Finally, in the office there are more or less the same number of modes both inside and outside, leading to an error quite similar to the reference error.

5.5.2 Model Identification Using Individual Models

The variability in the parameters shown in Figure 3.3 together with the sensitivity shown in Figure 5.9 suggest that it would be beneficial to use one individual model for each node instead of a common one for all the nodes. Using IMs is not a new idea. It has already been proposed indirectly for example in [Hightower et al., 2001, sec. 5], in which the author states that this '*insulates higher level ad-hoc location algorithms from hardware dependencies and the details of RSSI processing*'. The authors also come up with a calibration method '*to handle hardware variability and increase estimate accuracy*'. In [Savvides et al., 2001, sec. 3.1], the authors propose "*to identify each node against a reference node prior to deployment... so that the run-time RSSI measurements may be normalized to a common scale*", which in essence is equivalent to using IMs . In [Vallet et al., 2012] we demonstrated the advantages of using IMs using several types of polynomial models in the context of simultaneous localization and model identification. Here, Table 5.4a shows the errors that result when using a common log-normal model in contrast to using individual ones using the cross validation scheme described in Subsection 5.3.1.

The error reductions range from 20% in the office to 65% in the basketball field. Individual models incorporate more effects from the particular HW and nearby obstacles. A common model, in contrast, looses the information about the variability of parameters among different nodes, i.e., it ignores the dispersion of the curves from Figure 3.2 for n and z_0. To accommodate the differences among

nodes, the common reference model needs to increase its standard deviation. We can see this in the lower plot of Figure 3.2, where the values of σ for the common models (thick dots) are greater than the average of σ for individual ones in all environments. In terms of information, this essentially means that a collection of IMs contains more information than a single CM identified using all the data from all the nodes. As a result, the localization algorithm will give better position estimates when using individual models with respect to using a common one (see Table 5.4a).

It is worth noting that the improvement due to using IMs is largest in the basketball field. To understand the reasons behind this, we shall go back to the lower plot of Figure 3.3. We can see that the values of σ for the reference models (thick dots) are always greater than the mean of the distribution for individual models in all the environments. However, in the case of the basketball field, the thick dots are relatively further in the right tails of the distributions. Thus, the increase relative to the dispersion is the largest. That is: $(\sigma_{rm} - \bar{\sigma})/std(\sigma)$ is largest in the basketball field, where σ_{rm} is the standard deviation of the reference model and $\bar{\sigma}$ and $std(\sigma)$ are the mean and standard deviation of σ for the individual models. This means, in essence, that the gain of information when switching from common to individual models is relatively larger in the basketball field. The HW variability contributes the most to the dispersion of the model parameters in the basketball field with respect to the other environments. This suggests, again, that the use of individual models is especially effective addressing the HW variability problem in open spaces.

5.6 Generic Polynomial Models

The log-normal model was observed to predict well the observations from the basketball field. However, in some cases it was not able to explain properly the measurements from nodes in the other two more cluttered environments.

Figure 5.12 shows examples of sets of observations from two nodes located in the lobby and the office. These are representative examples that show typical tendencies of the RSS observed for cases in which the log-normal model was not capable of explaining well the observations. Panel 5.12a shows measurements from a node in the lobby. We can see that the curve is roughly divided into three parts: one with a rapid decrease of the RSS for short distances followed by another in the middle with more constant values (sometimes even increasing), and a final part with a final decay for the larger distances. The log-normal model is constantly decaying and cannot model this properly. Panel 5.12b shows a case for a node located in the office. The plot is in logarithmic scale, in which log-normally distributed data should follow a straight line. As we can see, the rate of decay of the RSS becomes larger as the distance increases, an issue that was

already discussed in Chapter 2.

Based on these observations, we decided to try using third-degree polynomial models on the distance and log-distance, which are more flexible than the log-normal model and can cope with the previous observations. The panels in Figure 5.12 show the fit of third-degree polynomials on the distance and log-distance. The subpanels on the right show the histograms of the residuals. These are also poorly modelled with Gaussians, which the log-normal model uses. We then decided to try using a MoG with two components, which can better accommodate skewed, multi-modal and heavy-tailed residuals.

For a generic polynomial of degree m with vector coefficients $\boldsymbol{p} = [p_1, \ldots, p_{\mathrm{m}+1}]^t$ and for a MoG with N components, the model for the observed RSS z for a given distance d can be written as

$$z = p_1\rho^m + p_2\rho^{m-1} + ... + p_{\mathrm{m}}\rho + p_{\mathrm{m}+1} + \xi = \boldsymbol{D}\boldsymbol{p} + \xi, \tag{5.6}$$

where $\boldsymbol{D} = [\rho^m, \rho^{m-1}, \ldots, \rho, 1]$, $\rho = d$ or $\rho = \log(d)$ depending on whether the polynomials depend on the distance or the log-distance respectively and $\xi \sim \sum_{n=1}^{N} \alpha_{\mathrm{n}}\zeta_{\mathrm{n}}$ represents a MoG. The MoG is characterized by the weighted sum of N normally distributed random variables $\zeta_{\mathrm{n}} \sim N(\mu_{\mathrm{n}}, \sigma_n^2)$ with mixture coefficients α_{n} restricted to $\sum_{n=1}^{N} \alpha_{\mathrm{n}} = 1$. The log-normal model is then a particular case for which $\rho = \log(d)$, $m = 1$, $N = 1$ and $\alpha_1 = 1$.

The likelihood of one observation under the model 5.6 is

$$p(z_{\mathrm{k}}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \sum_{\mathrm{n}=1}^{N} \alpha_{\mathrm{n}} \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left\{ -\frac{[z_{\mathrm{k}} - (\boldsymbol{D}_{\mathrm{k}}\boldsymbol{p} + \mu_{\mathrm{n}})]^2}{2\sigma_n^2} \right\}. \tag{5.7}$$

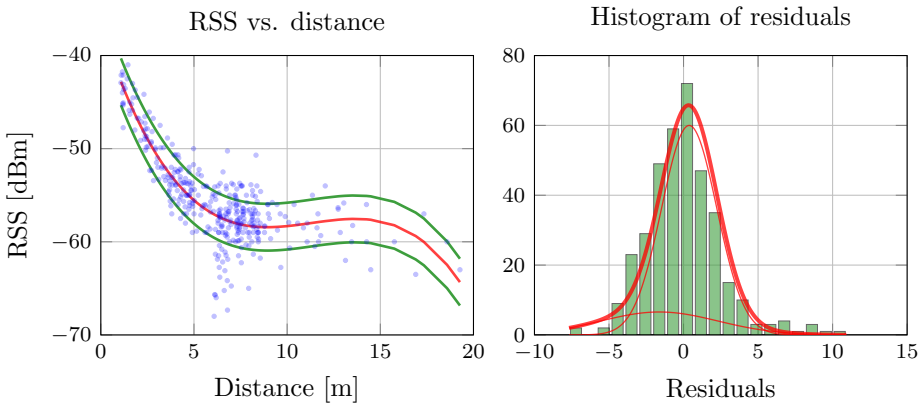
The model parameters are then $\boldsymbol{\theta} = [\boldsymbol{p}^t, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma}]$, with $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{\mathrm{N}}]$, $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_{\mathrm{N}}]$ and $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_{\mathrm{N}}]$. Assuming IID errors, the log-likelihood of a set of observations $\boldsymbol{z} = [z_1, z_2, ..., z_{\mathrm{K}}]$ is

$$L(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \ln \prod_{k=1}^{K} p(z_{\mathrm{k}}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \ln p(z_{\mathrm{k}}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}). \tag{5.8}$$

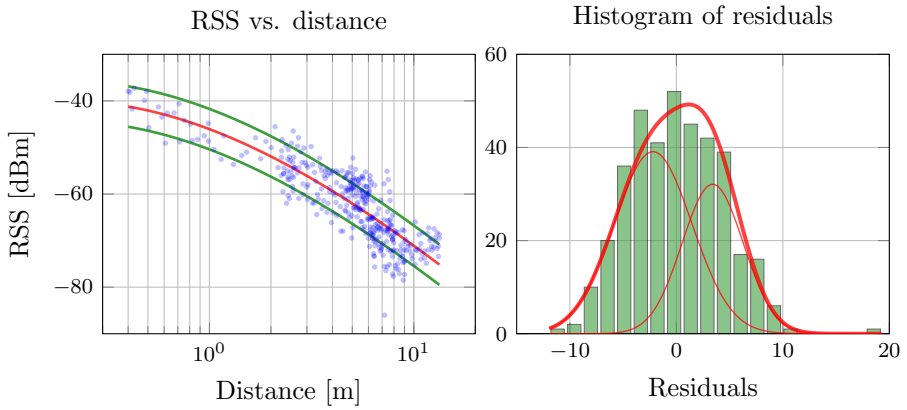
The ML position estimate is then

$$\hat{\vec{\boldsymbol{s}}} = \underset{\vec{\boldsymbol{s}}}{\operatorname{argmax}}\, L(\vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \underset{\vec{\boldsymbol{s}}}{\operatorname{argmax}} \sum_{i=1}^{N} \ln p(z_{\mathrm{k}}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}), \tag{5.9}$$

where the vector of model parameters $\boldsymbol{\theta}$ is supposedly known. In this thesis we estimate $\boldsymbol{p}$ using OLS, and $\boldsymbol{\alpha}$, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ using Expectation-Maximization [Dempster et al., 1977] to maximize the log-likelihood of the residuals.



(a) Lobby



(b) Office

Figure 5.12: Example of 3^{rd} degree polynomial models with MoGs. In the left fit of a P3M2N model for a node in the lobby. In the right fit of a P3LNM2N model for a node in the office. The red and green lines in the upper plots represent the mean and one standard deviation with respect to the mean respectively. In the lower plots the PDF of the MoGs is represented by the thick red lines, whereas the thin ones represent the weighted component distributions.

Table 5.6: Comparison of average errors of ML position estimates (distance in metres) when using different RSS-distance models in different environments. CM=common model, IM=individual model.

	P1LM1N			**P1LM2N**		
	CM	IM	Ratio(CM/IM)	CM	IM	Ratio(CM/IM)
Basketball field	0.94	0.33	2.83	0.97	0.35	2.77
Lobby	1.44	1.07	1.35	1.48	1.13	1.30
Office	0.97	0.78	1.25	0.97	0.79	1.24
	P3LM1N			**P3LM2N**		
	CM	IM	Ratio(CM/IM)	CM	IM	Ratio(CM/IM)
Basketball field	0.91	0.36	2.51	0.93	0.38	2.47
Lobby	1.40	1.88	0.75	1.42	1.89	0.75
Office	0.99	0.64	1.55	0.98	0.65	1.51
	P3M1N			**P3M2N**		
	CM	IM	Ratio(CM/IM)	CM	IM	Ratio(CM/IM)
Basketball field	0.93	0.36	2.57	0.94	0.36	2.62
Lobby	1.74	1.52	1.14	1.76	1.52	1.16
Office	1.00	0.63	1.60	1.01	0.64	1.59

We compared different models using as a metric the average localization error calculated as explained in Subsection 5.3.1. Table 5.6 shows the results for all the models considered. The notation is as follows. Models that depend on the distance are denoted for example P3M2N, to be read as *polynomial of* 3^{rd} *degree with a mixture of 2 normal components*. Models that depend on the logarithm of the distance are denoted for example P1LM2N, to be read as *polynomial of* 1^{st} *degree in the log-distance with a mixture of 2 normal components*. The log-normal model is thus denoted P1LM1N, as it is a first-order polynomial that depends on $\log(d)$ and has a single normal random variable. Thus, the results from the P1LM1N model are the same than those shown in Table 5.4a, which are reproduced in Table 5.6 for convenience.

Using individual models results, again, in a significant decrease of the average error in almost all scenarios and for all the models. Once more, the relative increase in the performance is largest in the basketball field. The errors in the lobby are the largest in general, and using individual models did not always prove to be better than using a common one. This can be explained by the presence of large dominant obstacles, which resulted in strong local environmental non-

inhomogeneity. This can affect the IMs strongly depending on the trajectory of the robot during the training phase. The IM can then become too specialized in explaining the training data, not being capable of explaining equally well the observations from the other trajectories. In other words: the model becomes over-fit.

Using MoGs to model the error does not necessarily result in a decrease of the average error. To the contrary, in almost all cases it increased slightly. This can again be the consequence of model over-fitting. Recall that the error measure of Equation (5.5) ensures that the data used for model identification are never used for localization. When using a MoG to model the error, the resulting model is fitted too precisely to the training data. When this model is used with other experimental data, it is not capable of adequately explaining well the differences between the two sets, and as a consequence the average localization error increases slightly.

Increasing the degree of the polynomial reduces the errors slightly in the office when using IMs, but not in other cases. This is somehow surprising, as one can naturally expect that the results should be at least equal. A third degree polynomial can be fit exactly to a first-degree polynomial, and has the advantage of being able to accommodate more generic shapes. The reason can be explained again with over-fitting. Precisely because of this extra flexibility of the third-order polynomial, it learns the particularities of the training data, and not just the general trend. When using this model for localization with a different dataset, the RSS predictions will then reflect these particularities, resulting in larger errors in the position estimates.

5.7 Conclusions

In this chapter we have studied the performance of several models and localization algorithms using real data from our experiments. Two types of methods have been explored, ML and LLS . The central method has been ML, due to its attractive well-known properties and the superior results predicted in the simulations of Chapter 4. Equivalently, the central model has been the log-normal model assuming IID errors, which is the most used RSS model in WSNs.

When using a robot as a mobile beacon and the log-normal model for localization, ML estimation suffers from two types of problems: estimator over-confidence and the presence of multiple characteristic local maxima. These problems are a direct consequence of the IID errors assumption and the use of a high density of measurements along the robot trajectory.

The over-confidence can be solved by artificially reducing the amount of information that each measurement carries by exponentiating its likelihood, which is a well-known method. One practical disadvantage of this solution is that it

requires manually selecting the exponent. We have done this upon inspection of the results, but in a real-world scenario this might be unrealistic. In later chapters we will assess two other methods, which we only briefly mention here: discarding correlated measurements and explicitly modelling the autocorrelation.

The presence of multiple maxima is a typical problem faced in estimation problems. One way to find the global maximum in an area of interest is to first use GS to calculate an initial position estimate, followed by the usage of a gradient-based solver such as L-BFGS-B. Using a fine enough grid guarantees that the global maximum can be found, but this can result in a large computational cost. The linear trajectory of the robot and the high spatial density of the measurements impact the shape of the log-likelihood in a characteristic way that can be exploited to aid gradient solvers to overcome local maxima. To this effect we have proposed the Maximize-Explore algorithm and shown that it can result in significant computational savings compared to a GS-LBFGSB approach.

The selection of the appropriate values for the model parameters (model identification) can significantly affect the performance of position estimators. In Chapter 3 we showed that there is a considerable variation of the model parameters depending on the particular nodes selected for the model identification, and in this chapter we have exposed the impact that this variability can have on the ML localization error. The relative sensitivity of ML localization is generally larger in places where the log-normal model explains well the observations (e.g. open environments). Additionally, we studied the effect of using data from more than one node to identify common models. This revealed that the main implication is not necessarily a direct and automatic reduction of the average error, but an increase in the robustness of the model in the sense that is less likely that it will result in large errors. Finally, we studied the possibility of using IMs. In our experiments this approach resulted in average error reductions of up to 65%. The main conclusions from this group of studies is then that the model identification process is especially relevant in WSNs, and different methods can lead to very different localization results. The HW variability and local inhomogeneity in the environment are the major sources of variability of the model parameters, and using individual models is a simple and relatively effective way to partially account for this variability.

Although the log-normal model is simple, attractive and widely used, it is not exempt from limitations, and it was experimentally clear that often it could not explain satisfactorily the RSS measurements, especially in the cluttered environments. Based on the observed behaviour of the RSS, we decided to generalize the model and try third degree polynomials to characterize the average signal and a mixture of Gaussians to model the error. We evaluated the improvements after each modification and concluded that the largest improvement is usually due to using IMs instead of CMs. When the environment is inhomogeneous, one

should be very cautious when using techniques that aim at learning the particularities of the local environment with over-simplistic radial models. Increasing the degree of the polynomial helps when there is a clear non-linear radial trend in the whole area of interest, like in highly homogeneous indoor setups, but it does not necessarily always help. Using MoGs is another technique to learn deeper particularities of the data, and has to be used with caution, as it can also lead to model over-fitting.

Finally, we validated the linear position estimators studied in Section 4.5 using experimental data and confirmed the exceptional performance of the GLS-ML and IGLS-ML variants in practical scenarios. The IGLS algorithm had to be modified to correct some occasional convergence issues encountered. Remarkably, after the modification, the IGLS-ML algorithm outperformed the traditional ML localization method in our experimental tests. All in all, the good performance of these methods opens the door to new lines of research, especially in computationally constrained applications.

Chapter 6

Node Localization Under Spatially Autocorrelated Noise

Contents

In previous chapters it was assumed that the error terms in the RSS log-normal model are IID. The reason for this was mathematical convenience, as it leads to rather simple mathematical expressions and easier to implement localization algorithms. In practice, however, in obstructed environments RSS measurements are spatially correlated, especially when they are collected in positions close to each other. That is the case in our experiments, in which the samples are drawn with a short spatial interval along the trajectory of the robot. This chapter is dedicated to localization methods that do not ignore this autocorrelation. Our aim is to devise techniques that take into account the presence of spatial autocorrelation and to compare them with those resulting from the IID assumption.

We start in Section 6.1 explaining what are the sources of the autocorrelation in RSS measurements under the log-normal model. We further explain the foundations of the autocorrelation model of our choice and review those found in the existing literature. Later, in Section 6.2, we specialize our literature review in studies that have used the log-normal model considering autocorrelated measurements for localization purposes.

In Chapter 5 we saw that ML position estimates were over-confident; we then argued that the cause of this is the IID errors assumption and the high spatial density of measurements. Section 6.3 shows this relation through the study of the CRLB in different illustrative cases. We will show how, in our case, taking into account the autocorrelation results in less confident position estimates.

Once the theoretical foundations have been set, Section 6.4 starts the empirical part of the chapter by showing correlograms calculated using our experimental data. The aim is to assess the adequacy of our choice of autocorrelation model. Later we explain in detail how the model parameters are estimated, and show their variability in different and within the same environment.

Once the models are properly identified, the rest of the chapter presents three different localization techniques that take into account the spatial autocorrelation of the residuals. The first two are based on ML estimation. Section 6.5 explores the possibility of using ML with a subset of the observations containing only uncorrelated samples. Alternatively, Section 6.6 shows the results of using full datasets, and thus exploiting all the information available from the experiments. In Section 6.7 the linear least square methods developed in Chapter 4 are extended to take into account the autocorrelation. Finally Section 6.8 presents the main conclusions of the chapter.

6.1 Spatial Autocorrelation in RSS Signals

6.1.1 Log-Normal Model Misspecification

Let us reproduce here for convenience the log-normal model for RSS measurements:

$$z = \bar{z} + \varepsilon \tag{6.1}$$

where

$$\bar{z} = z_0 - 10\, n \log\left(d/d_0\right) \tag{6.2}$$

$$\varepsilon \sim N(0, \sigma^2). \tag{6.3}$$

When can one assume that a set of measurements collected at spatially distributed points are uncorrelated under the log-normal model? Consider a T-R pair in free-space. The signals can travel straight from the transmitter to the receiver, and the only effect that it experiences is the attenuation due to the spreading of the energy of the waves. This attenuation is perfectly and deterministically predictable by the average $\bar{z}$, and the only source of uncertainty is the thermal noise, which is approximately white, Gaussian and entirely accounted for by the random variable ε. Thus, at least in free-space the RSS measurements are approximately uncorrelated.

Let us now introduce an obstacle into our ideal scenario. The waves now can pass through it, reflect, refract, diffract and scatter. Intuitively, in general we expect that the closer the transmitter and/or receiver are to the obstacle, the more noticeable its influence will be on the RSS. Moreover, we also expect that this influence will be similar at measurement points close to each other. This is the basis of the spatial autocorrelation: measurements collected in positions close to one another are affected by the same effects (obstacles in this case). Therefore, knowing about one measurement gives information about others around affected by the same spatial effects.

In a real scenario there is more than just one obstacle, and furthermore some of them might be mobile. Still, the idea is the same: the obstacles affect the RSS so that their effect is likely to be similar at close measurement points. In our particular experiments we have a mobile robot in three different rather static environments, each with a different degree of LoS. The robot sends packets whose RSS is measured by fixed receivers (nodes) at an effective rate of four measurements per linear metre traversed by the robot. Given the size of the obstacles, some of them much larger than one metre, it is quite reasonable to think that the measurements will not be independent of each other. Figure 6.1 illustrates this idea graphically. The RSS measurements associated with robot positions $\vec{\boldsymbol{r}}_{\mathrm{k}}$ and $\vec{\boldsymbol{r}}_{\mathrm{k+1}}$ are affected similarly by the presence of the obstacle (the grey hexagon).

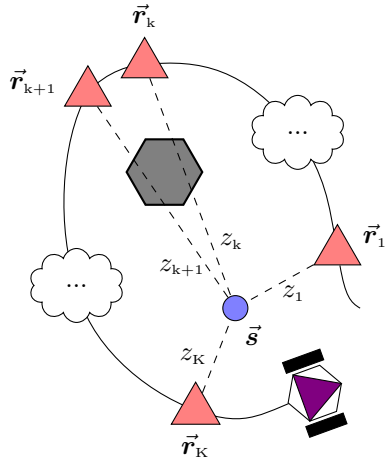


Figure 6.1: Spatial autocorrelation in the RSS due to the presence of an obstacle.

How is this correlation accounted for in the log-normal model? First of all, we have to recall that the log-normal model is rather simple. It consists of two terms: the (deterministic) average $\bar{z}$ that (given n and z_0) depends only on the T-R log-distance, and a random variable ε that accounts for *the rest*. When the model is identified using LSE or any other method, the parameters n and z_0 upon which the average depends are chosen so that the RSS predicted by $\bar{z}$ coincides as best as possible with the training dataset. Due to its radial dependency, the term $\bar{z}$ has a very limited capability of describing the effects of sources of inhomogeneity such as obstacles, for which more detailed geometrical information is needed. All the phenomena that $\bar{z}$ has not been able to explain satisfactorily goes implicitly to the residuals, that is, to ε. Therefore, the random variable ε accounts for the model misspecification in addition to any other source of uncertainty.

Returning to the example of free space, when the offending obstacle is added, $\bar{z}$ is no longer capable of fully explaining the observations. Suppose that we could somehow come up with a deterministic function $H = H(\vec{r}_{\mathrm{Tx}}, \vec{r}_{\mathrm{Rx}})$ which accurately models the effect of the obstacle on the RSS for any position of the transmitter $\vec{r}_{\mathrm{Tx}}$ and the receiver $\vec{r}_{\mathrm{Rx}}$. A faithful model that describes perfectly the RSS would then be

$$z = \bar{z} + H(\vec{r}_{\mathrm{Tx}}, \vec{r}_{\mathrm{Rx}}) + \varepsilon. \tag{6.4}$$

This model is not misspecified in that it explains accurately all the underlying physical phenomena. Moreover, because H accounts for the effects of the obstacle, $\bar{z}$ and ε still account only for the signal attenuation and thermal noise.

Let us now define $\mathcal{S}_{\mathcal{B}}$ as the functional space spanned by the base of functions $\mathcal{B} = \{1, \log(d)\}$ under a suitable scalar product. This subspace effectively covers all the possible first-degree polynomials on the log-distance. In particular, the coordinates of the log-normal model expressed in the base $\mathcal{B}$ are $(z_0, -10n)$. Let us decompose now

$$H(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}}) = H_{\parallel}(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}}) + H_{\perp}(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}}), \tag{6.5}$$

where $H_{\parallel}$ represents the projection of H onto $\mathcal{S}_{\mathcal{B}}$ and $H_{\perp}$ the orthogonal component. Then, 6.4 becomes

$$z = \bar{z} + H_{\parallel}(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}}) + H_{\perp}(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}}) + \varepsilon. \tag{6.6}$$

In reality the function H is not known, and the log-normal model has only two components: the average function, which lies in the subspace $\mathcal{S}_{\mathcal{B}}$, and *the rest*. In this sense, the reality applied to (6.6) means that the log-normal model becomes

$$z = \underbrace{\bar{z} + H_{\parallel}(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}})}_{\bar{z}'} + \underbrace{H_{\perp}(\vec{\boldsymbol{r}}_{\text{Tx}}, \vec{\boldsymbol{r}}_{\text{Rx}}) + \varepsilon}_{\varepsilon'} \tag{6.7}$$

$$= \bar{z}' + \varepsilon'. \tag{6.8}$$

Equation (6.8) represents our faithful model with the same functional shape as the log-normal model (6.1), and the transition from (6.7) to (6.8) shows the effect of the obstacle. Now the term accounting for the mean, $\bar{z}'$, includes the effects of the natural decay of the signal and the effects of the obstacle that can be represented by its functional shape. Additionally, the new random variable, ε', accounts for the thermal noise and for the effects of the obstacle that could not be incorporated into $\bar{z}'$. Whereas the original ε was independent of the position of the transmitter and receiver, ε' depends on them through $H_{\perp}$.

As we have just seen, because of the limited expressibility of the log-distance model's functional form, the presence of the obstacle has introduced a spatial dependence on the random variable ε'. In practice, accurately knowing the effects of a single obstacle in the RSS is very difficult, and even more so in a real dynamic environment with many obstructions. Therefore, H is not known, and for simplicity ε' is usually assumed to be random and independent of any position. In that case, the log-normal model is necessarily misspecified.

The lack of explanation for relevant factors then induces autocorrelation in the residuals. How problematic this is depends on the degree of misspecification, which is related to the homogeneity of the environment. In highly homogeneous environments H will, on average, be (radially) similar all around, and hence its component $H_{\parallel}$ will have more importance (norm) than $H_{\perp}$. Thus, the misspecification will have low relevance. As the environment becomes inhomogeneous

(e.g. because of the presence of dominant obstacles), H takes very (radially) different (function) values depending on the position of transmitter and receiver, and thus $H_\perp$ will acquire more relevance with respect to $H_\parallel$. The misspecification will be then more relevant.

Should we want to mitigate the effects of this misspecification, we have to add to the model some kind of spatial dependence of the RSS further than a simple log-distance relation. This can be done by allowing a spatial dependence of ε, as predicted in (6.7). We still do not know the details of this dependence, but, as we reasoned at the beginning of this section, we intuitively expect that these effects will be somehow similar in nearby positions. Recognizing the vagueness of this intuition, the alternative to trying to find the deterministic H, for which we do not have enough information, is again to embrace the uncertainty and use a probabilistic model that allows for more freedom and generality in its predictions. This idea of similarity between samples is translated into the probabilistic model as a correlation between samples, and their degree of similarity is dictated by the model used for the correlation. The spatial dependence enters then the model through the autocorrelation function. Considering the autocorrelation in this way then helps to mitigate, to a certain extent, the effects of the misspecification of the log-normal model.

Incorporating explicitly the autocorrelation adds further complexity to the probabilistic RSS models that we have been using thus far. The model is specified as the joint PDF $p(z_1, \ldots, z_K; \boldsymbol{\theta})$, where $\boldsymbol{z} = (z_1, \ldots, z_K)^t$ represents the vector of random observations associated with the robot positions $\boldsymbol{R} = (\vec{\boldsymbol{r}}_1, \ldots, \vec{\boldsymbol{r}}_K)$. The most direct implication of the IID assumption is that the joint PDF of all the measured dataset can be factorized as

$$p(z_1, \ldots, z_K; \boldsymbol{\theta}) = \prod_{k=1}^{K} p(z_k; \boldsymbol{\theta}). \tag{6.9}$$

This is no longer true when there is correlation among samples, and the problem is then how to formulate the joint density of the data so that the correlation is taken into account.

One tempting possibility is to consider the process $(z_1, \ldots, z_K)$ as a time-series, in which the measurements are an ordered sequence of random variables indexed by time. In our case the measurements would be indexed by the distance traversed by the robot between measurement points rather than by time. But there is a straight equivalence given that the measurement rate is constant along the robot trajectory. The fundamental problem of the time perspective is that the linear space traversed by the robot from one sample point to any another is not in general the distance between samples, which makes this approach very dependent on the trajectory of the robot. A more consistent approach is to consider a fully

spatial dependence of ε with respect to the position. This is the perspective used through the rest of this thesis.

6.1.2 Incorporating Spatial Autocorrelation

In a fully spatial approach the space itself is used as the index to order the data. This is an extension of the time series, and it constitutes what is known as a random field. [1] In our case, in which we have a discrete number of observations, a random field is a collection of random variables indexed by their position. The probabilistic model is still the joint PDF of all these variables, which still is to be defined completely.

Consider a region $\mathcal{R} \in \mathbb{R}^2$ with a node placed at position $\vec{\boldsymbol{s}} \in \mathcal{R}$ and a robot moving around sending packets. As usual, let us denote $\vec{\boldsymbol{r}}$ the position of the robot, where $\vec{\boldsymbol{r}} \in \mathcal{R}$. The packets received by the node will have an RSS which is a continuous random function dependent on the node's and robot's positions. Using the log-normal model as the basis for this function and indicating explicitly the (as yet formally unknown) spatial dependence of the random term we have:

$$z = \bar{z}(d, \boldsymbol{\theta}_{\mathrm{m}}) + \varepsilon(\vec{\boldsymbol{s}}, \vec{\boldsymbol{r}}), \;\; \varepsilon \sim \mathcal{N}(0, \sigma^2(\vec{\boldsymbol{s}}, \vec{\boldsymbol{r}})), \tag{6.10}$$

where $d = ||\vec{\boldsymbol{s}} - \vec{\boldsymbol{r}}||$ and $\boldsymbol{\theta}_{\mathrm{m}} = (-10n, z_0)^t$ is the vector of parameters upon which the mean depends. Equation (6.10) states that the marginal distribution of z is a Gaussian for any $\vec{\boldsymbol{s}}$, $\vec{\boldsymbol{r}}$ and $\boldsymbol{\theta}_{\mathrm{m}}$, and that its variance $\sigma^2(\vec{\boldsymbol{s}}, \vec{\boldsymbol{r}})$ is a function of the robot's and node's positions, function that still needs to be specified. It does not state anything about the relation among observations associated with different robot locations.

From the continuum of robot positions $\vec{\boldsymbol{r}}$, we now select a set of K points $\boldsymbol{R} = (\vec{\boldsymbol{r}}_1, \ldots, \vec{\boldsymbol{r}}_{\mathrm{K}})$ in which the robot sends packets and of which the node measures the corresponding vector of RSS samples $\boldsymbol{z} = (z_1, \ldots, z_{\mathrm{K}})^t$. Following the terminology used by geostatististicians, we will sometimes denote $\boldsymbol{R}$ as the set of control points and $\boldsymbol{z}$ as its associated vector of control values. We now assume that any collection $\boldsymbol{z}$ for any finite K has a joint multivariate normal distribution whose marginals are given by (6.10). That is:

$$\boldsymbol{z} = \bar{\boldsymbol{z}}(\boldsymbol{d}, \boldsymbol{\theta}_{\mathrm{m}}) + \boldsymbol{\varepsilon}(\vec{\boldsymbol{r}}, \vec{\boldsymbol{s}}), \; \boldsymbol{\varepsilon} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{R}, \vec{\boldsymbol{s}})\right), \tag{6.11}$$

where $\bar{\boldsymbol{z}} = (\bar{z}_1, \ldots, \bar{z}_{\mathrm{K}})^t$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_{\mathrm{K}})^t$ and $\boldsymbol{d} = (d_1, \ldots, d_{\mathrm{K}})^t$. Therefore, $\boldsymbol{z} \sim \mathcal{N}(\bar{\boldsymbol{z}}\left(\boldsymbol{d}, \boldsymbol{\theta}_{\mathrm{m}}\right), \boldsymbol{\Sigma}(\boldsymbol{R}, \vec{\boldsymbol{s}}))$. This effectively defines $\boldsymbol{\varepsilon}$ and $\boldsymbol{z}$ as Gaussian processes, which are completely determined by their respective mean and covariance. While their mean vectors are known ($\mathbf{0}$ and $\bar{\boldsymbol{z}}$ respectively), the covariance matrix of $\boldsymbol{\varepsilon}$

[1] For a detailed treatment of the extension from time series to random fields, the reader is referred to for example the excellent treatment in [Arbia, 2010, Ch.2].

still is not, and therefore neither that one of $\boldsymbol{z}$. Noting that $\mathrm{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$, we can write generically

$$\boldsymbol{\Sigma} = \mathrm{E}\left[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^t\right] = \begin{pmatrix} \mathrm{Cov}\left[\varepsilon_1, \varepsilon_1\right] & \dots & \mathrm{Cov}\left[\varepsilon_1, \varepsilon_\mathrm{K}\right] \\ \vdots & \ddots & \vdots \\ \mathrm{Cov}\left[\varepsilon_\mathrm{K}, \varepsilon_1\right] & \dots & \mathrm{Cov}\left[\varepsilon_\mathrm{K}, \varepsilon_\mathrm{K}\right] \end{pmatrix}. \tag{6.12}$$

Should we know the function $\sigma^2(\vec{\boldsymbol{s}}, \vec{\boldsymbol{r}}_\mathrm{k})$ of (6.10) (the variance of the marginals), we would also know the diagonal terms of (6.12) $\mathrm{Cov}\left[\varepsilon_\mathrm{k}, \varepsilon_\mathrm{k}\right] = \sigma^2(\vec{\boldsymbol{s}}, \vec{\boldsymbol{r}}_\mathrm{k})$. But we still have not specified it in any way.

The details of the spatial dependence of ε are not known, highly dependent on each particular setup and usually quite complex. It is then necessary to apply some restrictions to the autocorrelation model that we use, restrictions that simplify this dependence yet making it general enough so as to be applicable in different situations. In particular, it is customary to assume that ε is as a homogeneous, isotropic, wide-sense stationary and ergodic random field. A process is called stationary in the strict sense if its statistical properties are invariant to a shift in the origin [Papoulis and Pillai, 2002, sec. 9.1]. Stationary random fields are the equivalent to the stationary stochastic process, but now the shifts are in two dimensions, adding the possibility of translations and rotations. A homogeneous random field is random filed stationary under translations. An isotropic random field is random filed stationary under rotations. A homogeneous and isotropic random field is said to be stationary in the strict sense, and its PDF does not change under translations and rotations over the position [Arbia, 2010, Ch.2]. This implies that all univariate moments and all mixed moments of any order do not vary when the reference space is modified.

Another weaker form of stationarity is one in which only the mean and covariance are invariant. This is known as wide, weak or second-order stationarity. Given that a Gaussian process is defined completely by its first two moments, in our case stationarity in the strict and weak sense coincide [Arbia, 2010, Ch.2]. All in all, these restrictions imply that the covariances depend on the distances among control points, not on their particular positions. That is,

$$\mathrm{Cov}\left[\varepsilon_\mathrm{i}, \varepsilon_\mathrm{j}\right] = k(l_\mathrm{ij}),\ l_\mathrm{ij} = ||\vec{\boldsymbol{r}}_\mathrm{i} - \vec{\boldsymbol{r}}_\mathrm{j}||\,, \tag{6.13}$$

where $k(l)$ represents a covariance function that depends on the distance between the control points or lag[2] l. Clearly, the stationarity assumption limits the ability of the model to incorporate the details of the local spatial inhomogeneity, but it provides a significant simplification in its mathematical tractability. Applying (6.13) to (6.10) leads to $\sigma^2(\vec{\boldsymbol{s}}, \vec{\boldsymbol{r}}) = \sigma^2 = k(0)$. Finally, the ergodic property of our

[2]Following the nomenclature used by geo-statisticians.

second-order stationary random field implies that the spatial averages converge in probability to the set averages. This allows us to compute means and covariances from (large enough) sample realizations.

To characterize completely the RSS model, we still need to define the covariance function $k(l)$ with which $\mathbf{\Sigma}$ can be constructed. This function represents the probabilistic notion of similarity between observations at different points of space. In order for the PDF to be proper, the matrix $\mathbf{\Sigma}$ must be symmetric and positive-semidefinite. The symmetry is guaranteed by the construction of the matrix itself. Its positive semidefiniteness should be guaranteed by the selection of an appropriate function $k(l)$, which is, for the purpose, required to be positive semidefinite itself [Rasmussen and Williams, 2006, ch. 4].

As we argued in the previous subsection, the process ε incorporates all the effects that cannot be described by the log-distance mean. These can be roughly separated into spatial effects (mainly due to shadowing) and the effects of *any other source of uncertainty*. From this perspective, ε can be divided into

$$\begin{aligned} \varepsilon(l) &= \zeta(l) + \epsilon \\ \zeta &\sim \mathcal{N}(0, k_c(0)) \\ \epsilon &\sim \mathcal{N}(0, \sigma_n^2) \\ \mathrm{E}\left[\zeta\epsilon\right] &= 0, \end{aligned} \tag{6.14}$$

in which ζ contains all the spatial effects and ϵ the rest of the uncertainties.[3] Because we do not have any information about the effects accounted for by ϵ, we model it as white Gaussian noise with a constant variance σ_n^2 throughout $\mathcal{R}$. The independence assumption between ζ and ϵ is a reasonable one, as we have considered that the sources of their corresponding effects are different in nature. Comparing now (6.7) with the previous model (6.14), we see a clear similarity between ζ and $H_\perp$ in that they both account for the spatial effects. The fundamental difference is that, whereas $H_\perp$ is deterministic, accurate and of a great spatial complexity in the former, its counterpart ζ in the latter is a random field with a rather simple spatial dependence.

The most widely used autocovariance model in the RSS-related literature for the shadowing process (see Subsection 6.1.3) is an exponentially decaying model of the form

$$k_c(l) = \sigma_c^2 e^{-l/l_0}, \tag{6.15}$$

where the constant l_0 is the distance at which the correlation decreases to $1/e$, and it is usually referred to as the decorrelation distance. This leads to the

[3] This splitting of ε, as in (6.14), has already been proposed for RSS signals for example in [Patwari and Agrawal, 2008] and [Agrawal and Patwari, 2009].

following covariance function for ε in (6.14) (and therefore for z):

$$k(l, \boldsymbol{\theta}_{\mathrm{c}}) = \sigma_n^2 1_0(l) + \sigma_c^2 e^{-l/l_0}, \tag{6.16}$$

where $1_0(l)$ is the indicator function, which is equal to 1 only when $l = 0$ and zero otherwise, and $\boldsymbol{\theta}_{\mathrm{c}} = (\sigma_n, \sigma_c, l_0)^t$. This is effectively a valid covariance function, as it is the sum of two valid (positive definite) covariance functions [Rasmussen and Williams, 2006, Sec. 4.2.4].

Finally, using (6.11), (6.12) and (6.16), and denoting the set of distances between the control points $\boldsymbol{l} = \{l_{ij}, i = 1, \ldots, K, j = 1, \ldots, K\}$, the complete probabilistic model that we will use for the RSS including autocorrelation is:

$$\begin{aligned} \boldsymbol{z}(\boldsymbol{d}, \boldsymbol{\theta}) &= \bar{\boldsymbol{z}}(\boldsymbol{d}, \boldsymbol{\theta}_{\mathrm{m}}) + \boldsymbol{\varepsilon}(\boldsymbol{l}, \boldsymbol{\theta}_{\mathrm{c}}) \\ \bar{\boldsymbol{z}}(\boldsymbol{d}, \boldsymbol{\theta}_{\mathrm{m}}) &= \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}} \\ \boldsymbol{\varepsilon}(\boldsymbol{l}, \boldsymbol{\theta}_{\mathrm{c}}) &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{l}, \boldsymbol{\theta}_{\mathrm{c}})) \\ \boldsymbol{\Sigma}(\boldsymbol{l}, \boldsymbol{\theta}_{\mathrm{c}}) &= \sigma_n^2 \mathbf{I} + \sigma_c^2 \boldsymbol{W}, \end{aligned} \tag{6.17}$$

where

$$\boldsymbol{A} = \begin{pmatrix} \log(d_1/d_0) & 1 \\ \log(d_2/d_0) & 1 \\ \vdots & \vdots \\ \log(d_{\mathrm{K}}/d_0) & 1 \end{pmatrix}, \ \boldsymbol{\theta}_{\mathrm{m}} = \begin{pmatrix} -10n \\ z_0 \end{pmatrix}, \ \boldsymbol{\theta}_{\mathrm{c}} = \begin{pmatrix} \sigma_n \\ \sigma_c \\ l_0 \end{pmatrix}, \ \boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_{\mathrm{m}} \\ \boldsymbol{\theta}_{\mathrm{c}} \end{pmatrix}, \tag{6.18}$$

$$\boldsymbol{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,\mathrm{K}} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,\mathrm{K}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{\mathrm{K},1} & w_{\mathrm{K},2} & \cdots & w_{\mathrm{K},\mathrm{K}} \end{pmatrix}, \ w_{\mathrm{i,j}} = e^{-l_{\mathrm{i,j}}/l_0}, \ l_{\mathrm{i,j}} = \|\vec{\boldsymbol{r}}_{\mathrm{i}} - \vec{\boldsymbol{r}}_{\mathrm{j}}\|. \tag{6.19}$$

6.1.3 RSS Autocorrelation Models in the Literature

The notion of probabilistic similarity provided by the autocorrelation function has been used in different disciplines, such as spatial econometrics, geostatistics and machine learning. Each discipline has developed autocorrelation models that best fit their own purpose, and therefore numerous models exist that one can choose from (see e.g. [LeSage and Pace, 2009, ch. 2], [Cressie, 1993, ch. 2], [Rasmussen and Williams, 2006, ch. 4]).

In the RSS-related literature, the correlation due to shadow fading has been studied empirically since the early 1990s in different environments and at different frequencies. The main motivation for studying the autocorrelation has to do with the oversimplicity of the majority of existing propagation models, which can lead

to misleading results from simulations and analysis [Kotz et al., 2003]. Most of the research on autocorrelation has been driven by the cellular technology, and therefore the studies have been carried out mostly in different types of urban scenarios [Marsan et al., 1990, Gudmundson, 1991, Malmgren, 1997, Algans et al., 2002, Weitzen and Lowe, 2002]. Indoor environments have also been characterized [Liberti and Rappaport, 1992, Turin et al., 2002, Jalden et al., 2007], although they have received comparatively less attention.

The typical way of acquiring measurements is to move a transmitter/receiver along predefined trajectories (roads/corridors). Knowing the speed of the mobile platform and the sampling rate, it is possible to know the distance between sampling points. After gathering the data, the trend and fast fading are removed from the samples prior to the correlation analysis, which is typically done using traditional time-based methods. In contrast, in this thesis we will follow a fully spatial approach.

Different autocorrelation models have been proposed based on field measurements. [Szyszkowicz et al., 2010] groups them in five categories in a detailed survey: a) a constant model, which assigns a constant correlation in all positions of space; b) absolute distance-only models, in which the correlation depends only on the T-R distance; c) angle-only models, in which the correlation is a function of the angle of arrival; d) separable models, which are constructed by multiplying 1D models (e.g. distance, angle and distance-ratio R (see below)); and e) more elaborate (non-separable) models. Besides giving an exhaustive review of the models used in the field, the authors analyze in detail their mathematical feasibility and evaluate their physical plausibility. Some of the models were demonstrated to not to guarantee feasibility in all circumstances, and the authors encourage the community to use only those that ensure, by design, the positive definiteness of the covariance matrix.

Perhaps the simplest model is the constant correlation model, which assigns the same correlation to all of the links. That is:

$$h(l) = \rho, \tag{6.20}$$

where ρ is the correlation coefficient. This model is positive definite for $0 \leq \rho < 1$, but not for $-1 \leq \rho < 0$ [Szyszkowicz et al., 2010].

The most common model for the autocorrelation between two RSS samples is an exponentially decaying function of the distance between control points, of the form

$$h(l) = e^{-l/l_0}, \tag{6.21}$$

model that we used for the autocovariance of ζ in (6.15). This can be seen as a multidimensional extension of the Orstein-Uhlenbeck process [Uhlenbeck

and Ornstein, 1930]. Although the model is usually attributed to Gudmundson,[Gudmundson, 1991] [4] Marsan *et al.* also proposed in essence the same model earlier in [Marsan et al., 1990].

Other more complex variations using exponentials have also been proposed. For example, in [Mawira, 1992], [Sørensen, 1998] and [Algans et al., 2002] the authors use a linear combination of two independent decaying exponentials in their empirical studies in urban macro cells. The model is of the form

$$h(l) = \alpha e^{-l/l_1} + (1-\alpha)e^{-l/l_2},$$

with $0 < \alpha < 1$. This model corresponds to a process determined by the addition of two independent random variables with their own exponentially decaying autocorrelation. With respect to the simple exponential model, it has the advantage of allowing different decay rates for two ranges of proximity (e.g. an initial fast decay within short distances and a slow decay for long ones, as measured in [Sørensen, 1998]). Other variations used are the squared exponential $\exp\{\frac{-l^2}{2l_0^2}\}$ [Graziosi, 1999, Mandayam et al., 1996, Ferris et al., 2006] and the $\gamma-$exponential $\exp\{-(l/l_0)^\gamma\}$ [Patwari and Hero, 2004, Catrein and Mathar, 2008].[5]

All of these models are positive definite, but depend only on the distance between control points l. This has been argued to be non physically plausible in [Szyszkowicz et al., 2010], in which the authors give some criteria that, in their opinion, spatial correlation functions for RSS should comply with besides being mathematically feasible. In essence the model should not depend only on l, but also on the relative angle between the control points $\theta \in [0 \leq \theta \leq \pi]$ (as seen by the base station) and the ratio $R = |10\log(d_1/d_2)|$, where d_1 and d_2 are the distances between the base station (node) and the two terminals (control points). It should also be a continuous, smooth, non-negative and non-increasing function of l, θ and R. Additionally, for small values of l and θ its value should be close to one, and far from it when l is large. These criteria are the result of considering the shadowing as a large-scale phenomenon that can be explained by the partial overlap of the propagation medium. Szyszkowicz *et al.* proposed the following

[4] The original formulation in [Gudmundson, 1991] is $A(l) = \rho^{\delta/D}$, where ρ is the correlation between two samples separated a distance D. This expression can be alternatively formulated as in (6.15) by letting $\rho = 1/e$ and $D = l_0$.

[5] In [Patwari and Hero, 2004] the value γ=1 is used, and the model is then the simple decaying exponential. In [Catrein and Mathar, 2008], it can be shown that the model used is effectively the $\gamma-$exponential using $\theta_1 = e^{-l_0{}^\gamma}$.

model as the best of those reviewed [6]

$$\begin{aligned} h(\theta, R) &= h_{\phi}(\theta) h_{\mathrm{R}}(R) \\ h_{\phi}(\theta) &= \begin{cases} a - (a-b)\theta/\theta_0, & \theta \leq \theta_0 \\ b, & \theta > \theta_0 \end{cases} \\ h_{\mathrm{R}}(R) &= \max(0, 1 - R/R_0), \end{aligned} \tag{6.22}$$

with $0 \leq b \leq a \leq 1$, and $0 \leq \theta_0 \leq \pi$.

Although we agree in that (6.22) is in principle more plausible than a simple univariate exponential model (or a combination of $1_0(l)$ plus an exponential), for our localization purposes it poses an important challenge. As usual, we will use ML to estimate $\vec{\boldsymbol{s}}$ and a gradient-based optimizer, and therefore it will be necessary to compute the log-likelihood in many iterations and for different values of $\vec{\boldsymbol{s}}$. The log-likelihood in turn depends on $\boldsymbol{\Sigma}^{-1}$, and using (6.22) causes $\boldsymbol{\Sigma}$ to depend on $\vec{\boldsymbol{s}}$ through d_1 and d_2 (see the definition for R above). Therefore, each evaluation of the log-likelihood for different values of $\vec{\boldsymbol{s}}$ requires inverting $\boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}$ with K in the order of 10^3. This is not the case when using the chosen model (6.17), in which $\boldsymbol{\Sigma}$ depends on the fixed and known distances between the control points, and therefore it needs to be inverted only once. Therefore, using (6.22) adds significantly more computing requirements. For the sake of simplicity and lower computation demand, we decide to use the autocovariance function of (6.16).

The autocorrelation models studied thus far are valid for link pairs with a common endpoint. As Agrawal and Patwari proved, when considering arbitrary links in a multi-hop network, the exponentially decaying autocorrelation model of (6.21) cannot be extended directly, as it can result in a non-positive semi-definite covariance matrix [Agrawal and Patwari, 2009]. For such type of networks and for arbitrary links, they propose to model the shadowing losses as a line integral over an isotropic and homogeneous wide-sense stationary zero mean Gaussian spatial field representing the losses at each point of the space. The correlation between arbitrary links is then taken into account via the integration over the same field. They refer to this model as the network shadowing model (NeSh). As the authors point out, the use of attenuation factors in propagation models to account for the shadowing losses, such as in [Seidel and Rappaport, 1992] and [Durgin et al., 1998], is essentially a line integral over a spatially dependent function. The difference is that in the NeSh model the spatial field is of random nature, whereas when using attenuation factors the line integrals are substituted by summations over the attenuation of particular objects that block the T-R LoS.

[6] The model is in fact a modification based on the work in [Klingenbrunn and Mogensen, 1999], and is the only one that they found to be both positive definite and complying with their feasibility criteria.

6.2 Localization Using the Log-Normal Model

In Chapter 5 we presented localization results from methods using the log-normal model and assuming IID measurements. In this chapter we will present results for the same methods, this time explicitly modelling the spatial autocorrelation. We are then interested in the results from the literature regarding the effects of the autocorrelation under the log-normal model, and more specifically on the effects of ignoring it.

The publications reviewed within this context can be divided attending to the autocorrelation model used. We found references using constant, exponentially decaying and NeSh models. To our surprise we did not find the model (6.17) that we use in this thesis.

All the publications reviewed use simulations that follow a similar scheme: a) layout generation, in which the nodes to be located are deployed in uniformly random positions within a squared or circular area; b) placement of the beacons or reference nodes, either in selected positions with good geometry, such as the borders of the domain, or in random positions; c) generation of correlated data using the corresponding model; and d) application of the methods under study using the artificial data.

The constant correlation model has been used for example in [Lee and Buehrer, 2009], [Yang and Chen, 2009], [Mailaender, 2011], [Mailaender, 2012], and [Vaghefi and Buehrer, 2013]. Mailaender uses a constant autocorrelation model in [Mailaender, 2012], for which he presents the CRLB in [Mailaender, 2011]. Interestingly, under the log-normal model with constant correlation this bound does not depend on the correlation coefficient, demonstrating that an optimal estimator does not need this knowledge. Mailaender opted to pose the localization problem using the LS criterion leading to a non-linear problem which solves using different minimizers. He recommended using Levenverg-Marquardt on the basis of a higher convergence rate.

[Lee and Buehrer, 2009] propose to use differential received signal strength (DRSS) measurements instead of simple RSS. Using DRSS has the advantage of eliminating the dependency of the localization algorithm on the parameter z_0 of the log-normal model (see (6.2)). The authors formulate the position estimators using LS criterion and study via simulations results for different levels of correlation and number of anchor nodes. When using DRSS, the root mean squared error (RMSE) decreases as the autocorrelation coefficient increases for all number of anchors used (4 to 16). Using simple RSS results in higher RMSE for smaller correlation coefficients when using less than a certain number of anchors (9 in their case), and higher for larger amounts of reference nodes. Overall, DRSS performs clearly better than RSS only with high correlation values (0.8).

In [Vaghefi and Buehrer, 2013] the authors transform the non-linear ML formulation of the problem into a linear one using semi-definite programming (SDP), which required the relaxation of the cost function into a convex surface. The solution can then be found using efficient standard SDP methods.[7] They then compare their proposed method against others in a simulated environment consisting of four anchor nodes with fixed location[8] and 121 nodes placed uniformly in fixed locations in a 10x10m^2 area. According to their simulations, the presence of spatial autocorrelation can result in more accurate position estimates. Additionally, assuming IID measurements when they are correlated results in an increase of the average error of ML position estimators.

The exponentially decaying autocorrelation model has been used for example in [Yang and Tsai, 2009] and [Flåm et al., 2010]. In [Yang and Tsai, 2009] the authors used a Kalman filter to track the shadowing process, followed by a ML step for the position estimation using the path loss with the shadowing removed. They simulated a mobile device following a straight line with constant speed. They showed that the shadowing can be successfully tracked by the Kalman filter, and that the estimated and real locations converge after some time. Unfortunately, they did not provide a comparison between the results when considering IID and autocorrelated samples.

[Flåm et al., 2010] proposes a minimum mean square error (MMSE) Bayesian position estimator using a uniform prior for the position to be estimated represented by a set of position candidates. The resulting position estimator (the mean of the posterior) is a weighted sum of the position candidates, where the weights are the individual normalized likelihoods. The authors call this method weighted likelihood (WL). Their study compares via simulations the RMSE of several algorithms: an (GS-initialized) ML, WL and WL ignoring the autocorrelation, always discarding the weakest RSS measurements. The simulated environment consists on a circular cell with 100 m radius, a prior with 100 candidate source positions located in a grid and 10 reference nodes with a known, uniformly distributed position. In this scenario, they located 50 nodes whose position is estimated using 300 combinations of the $N = 10$ uniformly randomly placed reference nodes. For the grid-search ML, they used a grid with 10 m separation. They used a decorrelation distance of 80 m.[9] The results for their simulation setup show an increase of the RMSE when ignoring the autocorrelation.

[7] This technique has already been used in [Biswas and Ye, 2004].

[8] The position of the anchor nodes is not specified.

[9] When considering a 100 m radius cell and randomly placed nodes and references, the average distance is of about 47.5 m (calculated numerically using 10^4 realizations), leading to an average correlation coefficient of 0.55.

The NeSh model in the context of localization is analyzed by its authors in [Patwari and Agrawal, 2008], where the CRLB for position estimators is calculated and compared to the CRLB using uncorrelated measurements. The study uses simulations with nodes placed in an area of $4x4m^2$ and with rather good geometry of the problem (reference nodes located in the corners). In their study the authors found that using autocorrelation information decreased the location estimation variance bound (the trace of the CRLB) by 4.5% in average for a random deployment with 4 anchor nodes and 12 with unknown position. The authors analyze these results in terms of the contributions to the FIM of the mean and covariance terms of the measurement model (see (4.40)). They show how, when the autocorrelation is present, the contribution from the mean term decreases the information, while the contribution from the covariance term increases it, leading to a net moderate increase on average.

[Al-Dhalaan and Lambadaris, 2010] use the NeSh correlation model and an SDP-relaxation technique for the localization. Their simulation setup is a 5mx5m area with a single realization containing 60 sensors and 25 anchors, which both seem to be uniformly distributed.[10] For this setup the authors present the RMSE as a function of σ for several values of the average maximum communication range for the radios. Using their algorithm, the RMSE increases when ignoring the autocorrelation.

6.3 Overconfidence

In Section 5.1 we explained two characteristic problems that ML position estimator suffered under the log-normal model with IID noise and a high spatial density of measurements. One of them is the over-confidence of the estimates, which we solved by artificially reducing the amount of information contained in each sample substituting $p(z|s,\theta)$ by $p(z|s,\theta)^\alpha$ with $0 < \alpha < 1$. We then argued that this method has the practical inconvenience of needing to manually choose the coefficient α. In this chapter we aim at solving the overconfidence problem through specifically incorporating the spatial autocorrelation into the model.

A reduction of the overconfidence is equivalent to an increase of the estimator's covariance, which is what we are trying to achieve by incorporating the autocorrelation into the RSS model. This somehow contrasts with the general trend observed in the literature review from the previous section, which suggested that the RMSE increases when the autocorrelation is ignored.

In this section we present a theoretical study that demonstrates that accounting for the correlation does not automatically imply a decrease neither an increase in the minimum covariance of unbiased position estimators, and that the resulting

[10] This is not stated explicitly.

effect depends on the geometry of the problem and the shape of the autocorrelation function. In the experimental conditions of this thesis, acknowledging the autocorrelation leads in general to an increase of the covariance, which mitigates the over-confidence problem. The study that follows in this section, then, does not include the effect of the bias on the RMSE, and concentrates only on the covariance.

One view regarding the autocorrelation is that having one sample available will give us information about other samples taken in close positions where the autocorrelation is still significant. Equivalently, the availability of new samples will add less new information compared to considering them IID. Therefore, given a set of samples, the IID assumption implies that the total amount of information contained in the dataset is larger that when considering them correlated. And, should this correlation exist, this extra information will make the estimators overconfident. In our case, this is emphasized by the fact that the samples are associated with robot positions in close proximity to one another.

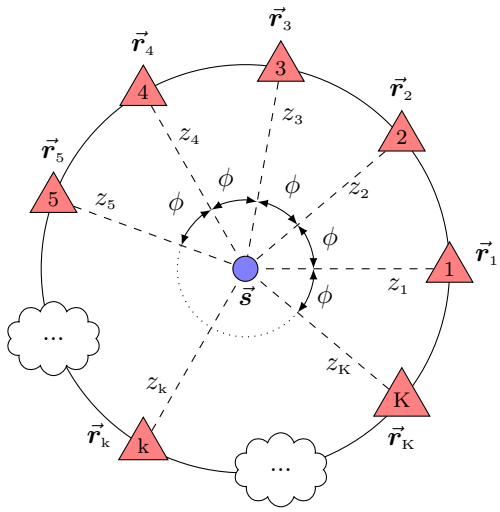
To illustrate and study this effect, consider the scenario depicted in Panel 6.2a. We have K beacons located in a circle with equiangular separation, $\phi = \frac{2\pi}{K}$rad, with positions denoted as $\vec{\boldsymbol{r}}_{\mathrm{k}}$, $k = \{1, \ldots, K\}$. Each beacon sends one packet whose RSS z_k is measured by a node located in the centre, whose position $\vec{\boldsymbol{s}}$ we want to estimate. In order to simulate a real scenario with spatially correlated RSS, we assume that $\boldsymbol{z} = (z_1, \ldots, z_K)^t$ is a process that strictly follows our theoretical model of (6.17). In the following we will refer to this process as the *generating process*.

The previous scenario is an illustrative simplified representation of what happens in our experiments: we want to locate a node that receives beacons sent by a mobile robot with a certain spatial periodicity. Our aim is to study first what happens to the covariance matrix of position estimators $\hat{\vec{\boldsymbol{s}}}$ when, being present, the spatial correlation is ignored, and second how the beacon density affects this covariance. We will do this by comparing the position estimators that use two different models: the one of the generating process, which simulates the reality, and another that models the generating process but assuming IID samples. In the following, we will refer to the latter one as the *IID model*.

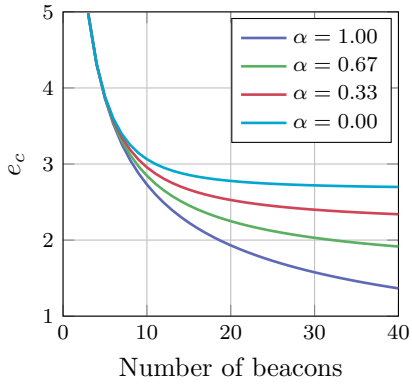
Clearly, the variance of the residuals of our generating process under the IID model is $\sigma^2 = \sigma_n^2 + \sigma_c^2$. Using this we can write $\sigma_n^2 = \alpha\sigma^2$ and $\sigma_c^2 = (1-\alpha)\sigma^2$, where $0 \leq \alpha \leq 1$. The expression of $\boldsymbol{\Sigma}$ in (6.17) can then be reformulated as

$$\boldsymbol{\Sigma} = \sigma_n^2\mathbf{I} + \sigma_c^2\boldsymbol{W} = \alpha\sigma^2\mathbf{I} + (1-\alpha)\sigma^2\boldsymbol{W} = \sigma^2\left[\alpha\mathbf{I} + (1-\alpha)\boldsymbol{W}\right]. \tag{6.23}$$

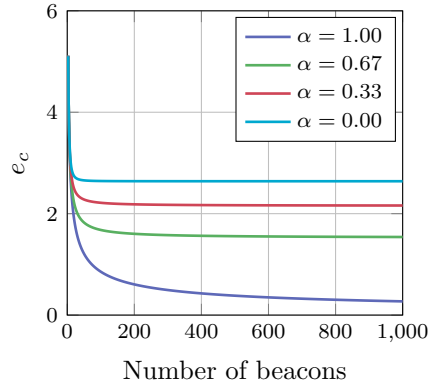
With this alternative parameterization, any of our (autocorrelated) generating processes will necessarily have $1 < \alpha \leq 0$. Furthermore, we can compute the covariance matrix of its associated IID model by simply selecting $\alpha = 1$ keeping the same value of σ. Thus, studying the effect of ignoring the autocorrelation



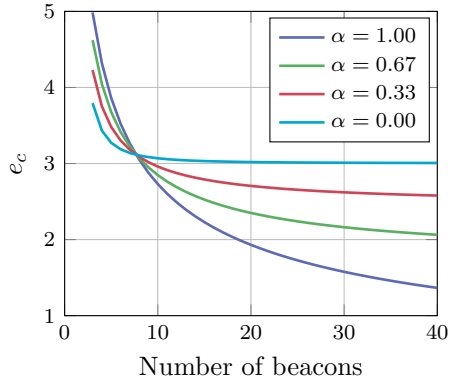
(a) Simulation scenario with K beacons.



(b) $l_0 = 5$ m, $3 \leq K \leq 40$.



(c) $l_0 = 5$ m, $3 \leq K \leq 1000$.



(d) $l_0 = 30$ m.

Figure 6.2: Simulation scenario used to study the effects of the measurements' spatial density and resulting characteristic error as a function of number of beacons.

of the generating processes can be done by comparing the estimators under two different models: one equal to the generating process, with specific values of σ and α, and its associated IID version, with the same value of σ and setting $\alpha = 1$. In fact, note that, for any fixed value of σ, there are many (virtually infinite) generating processes associated with the same IID model: any with $\alpha \neq 1$.

Comparing position estimators using different models can be done by calculating the CRLB associated with the underlying models. This will give us the minimum variance achievable by any unbiased position estimator that uses these models. The FIM for our model (6.17) is given by the expression (4.40), which simplifies to[11]

$$\mathbf{FIM}(\vec{s}, \boldsymbol{\theta}) = \left(\frac{10n}{\ln 10}\right)^2 [\boldsymbol{D}^{-2}(\mathbf{1}\vec{s}^{\,t} - \boldsymbol{R})]^t \boldsymbol{\Sigma}^{-1} \boldsymbol{D}^{-2}(\mathbf{1}\vec{s}^{\,t} - \boldsymbol{R}), \tag{6.24}$$

where (6.25)

$$\mathbf{1} = (1, \ldots, 1)^t, \ \dim(\mathbf{1}) = K \times 1 \tag{6.26}$$

$$\boldsymbol{R} = (\vec{r}, \ldots, \vec{r}_{\text{K}})^t, \ \dim(\boldsymbol{R}) = K \times 2 \tag{6.27}$$

$$\boldsymbol{D} = \text{diag}(d_1, \ldots, d_K), \ \dim(\boldsymbol{D}) = K \times K, \tag{6.28}$$

and $\boldsymbol{\Sigma}$ is given by (6.23), which now depends on the parameters σ, α and l_0. The CRLB is then the inverse of the FIM.

Panels 6.2b and 6.2c represent the characteristic error $e_c = \sqrt{\text{Tr}\,(\mathbf{CRLB})}$ for the situation depicted in Panel 6.2a, with $d = 15m$, $n = 4$, $z_0 = -70$, $\sigma = 5$ and $l_0 = 5$. The characteristic error is plotted as a function of the number of beacons, K, for different generating processes (different values of $\alpha \neq 1$) and their associated IID model ($\alpha = 1$). Panel 6.2b presents the results for $3 \leq K \leq 40$, and Panel 6.2c for $3 \leq K \leq 1000$.

The characteristic error is minimum when $\alpha = 1$ for any $K \geq 5$, indicating that using the IID model can result in covariances of $\hat{\vec{s}}$ lower that the CRLB under the model of the generating process. In other words, considering IID measurements in the presence of spatial correlation will result in overconfident position estimates. This overconfidence increases with the number of beacons, that is, with the spatial measurements' density. Equivalently, taking into account the autocorrelation in the model increases the variance of the position estimate with respect to assuming IID measurements, solving the overconfidence problem.

Two points deserve further comment. First, it is interesting to see that e_c seems to posses a lower limit as the number of beacons increases. For uncorrelated processes e_c seems to approach 0 for $K \to \infty$, which means that, in theory, one could achieve an arbitrarily high precision by adding a large enough number of

[11] Calculated for $\vec{s}$. Note then that, because $\boldsymbol{\Sigma}_{\mathbf{L}}$ does not depend on $\vec{s}$, the second line of (6.17) becomes zero.

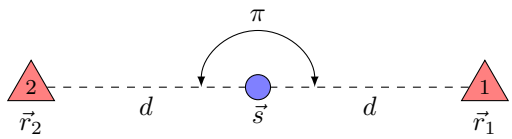


Figure 6.3: Simulation scenario with 2 beacons.

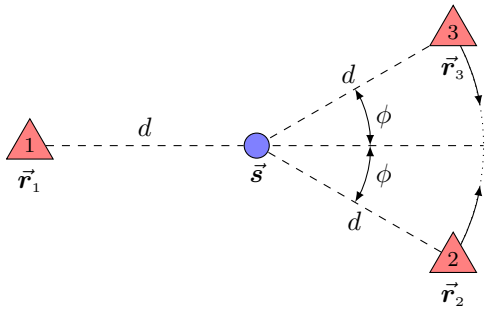
beacons. For processes in which $\alpha \neq 1$, the lower limit seems to be larger, being largest when $\alpha = 0$ (pure exponentially decaying correlation). We do not prove this mathematically, but simply provide the results of simulations for K up to 1000 (see Panel 6.2c). This suggests that, for autocorrelated processes under our model, it is not possible to obtain arbitrarily precise position estimators by simply adding more beacons. In practice, this can help decide the largest number of beacons that is worth installing in a real system. Equivalently, in cases when a mobile beacon is used for localization, such as in this thesis, this can help to decide the spatial sampling rate to be used for an effective use of the data. This limit depends on the shape of the correlation function, and therefore on the values of σ, α and l_0.

Second, although it cannot be seen clearly in panels 6.2b and 6.2c, the characteristic error is largest for the IID model when $K < 5$, which is contrary to our intuition that the IID model is always more informative. This is made more clear in Panel 6.2d, which is generated with a much larger $l_0 = 30$. In this case, the IID model did not become the most informative until $K \geq 8$. Therefore, whether or not ignoring the autocorrelation will result in a smaller or larger estimator's covariance depends, first, on the shape of the autocorrelation function, and, second, on the geometrical configuration of the problem. One thus cannot make general claims for arbitrary autocorrelation functions and problem geometries.

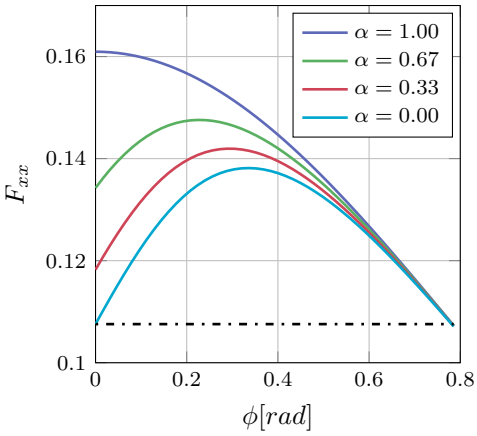
One simple sample case for which the IID model will always be the least informative is the one represented in Figure 6.3, for which it is easy to calculate the FIM analytically. It can expressed as (see Appendix B.1 for a detailed derivation):

$$\mathbf{FIM} = \frac{2}{d^2(1-a)} \left(\frac{10n}{\sigma \ln 10} \right)^2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \text{ where } a = (1-\alpha)e^{-2d/l_0}. \tag{6.29}$$

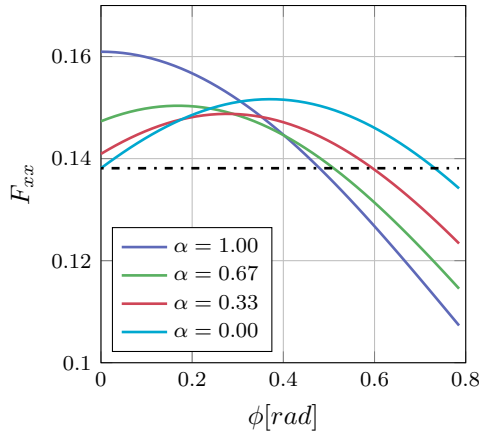
Clearly $0 \leq a \leq 1$, and the information about the x component will be minimum when $a = 0$, which corresponds to $\alpha = 1$. In other words: in this configuration the IID model is the least informative always under our generative model. Note that the information about the coordinate y is zero, and therefore the FIM is not invertible in this particular case.



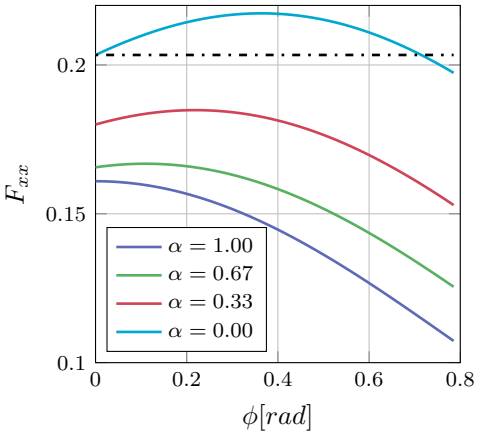
(a) Simulation scenario with 3 beacons.



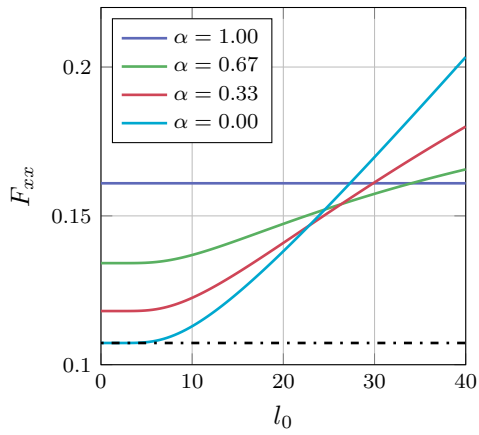
(b) $l_0 = 5m$.



(c) $l_0 = 20m$.



(d) $l_0 = 40m$.



(e) $\phi = 0$, $0 \leq l_0 \leq 40m$.

Figure 6.4: Simulation scenario with 3 beacons and its corresponding FIM.

Another illustrative setup is the one presented in Panel 6.4a. In this case we have three beacons in a horizontally symmetric configuration equidistant from a node located at the origin of coordinates ($\vec{\boldsymbol{s}} = (0,0)^t$). The first beacon lays in a fixed position on the x axis, $\vec{\boldsymbol{r}}_1 = (-d,0)^t$, while beacons two and three are located in positions $\vec{\boldsymbol{r}}_2 = (d\angle -\phi)$ and $\vec{\boldsymbol{r}}_3 = (d\angle\phi)$ in polar coordinates, respectively. We want to study what happens with the information available to position estimators $\hat{\vec{\boldsymbol{s}}}$ as the beacons two and three become closer, that is, as $\phi \to 0$.

The FIM for this particular deployment can be calculated analytically, and can be expressed as (see Appendix B.2 for derivation):

$$\mathbf{FIM} = \begin{pmatrix} F_{xx} & 0 \\ 0 & F_{yy} \end{pmatrix}$$

where

$$\begin{aligned} F_{xx} &= \left(\frac{10n}{d\sigma \ln 10}\right)^2 \frac{1 + b + 4a\cos(\phi) + 2\cos^2(\phi)}{b - 2a^2 + 1} \\ F_{yy} &= \left(\frac{10n}{d\sigma \ln 10}\right)^2 \frac{2\sin^2(\phi)}{1 - b} \\ a &= (1-\alpha)\exp\left\{-\frac{2d\,|\cos(\phi/2)|}{l_0}\right\} \\ b &= (1-\alpha)\exp\left\{-\frac{2d\,|\sin(\phi)|}{l_0}\right\}. \end{aligned} \tag{6.30}$$

Panels 6.4b, 6.4c and 6.4d plot F_{xx} as a function of ϕ for $l_0 = 5$, $l_0 = 20$ and $l_0 = 40$ m, respectively, and for different values of α. Panel 6.4b shows that, with $l_0 = 5m$, F_{xx} is maximum when $\alpha = 1$ for the entire range of ϕ represented and minimum when $\alpha = 0$. For values of $\alpha \neq 1$, the tendency is a reduction of the amount of information as beacons two and three become closer ($\phi \to 0$). Furthermore, the difference between values of F_{xx} increases for different values of α. This means again that using the IID model results in overconfident position estimators in the x coordinate, and this overconfidence will be higher the closer the beacons are.

For a much larger $l_0 = 20m$, F_{xx} is not necessarily maximum anymore when $\alpha = 1$ for all ϕ (see Figure 6.4c). Eventually, for a large enough l_0, the IID model can become the least informative for the entire range of ϕ represented (see Figure 6.4d). This demonstrates once more that whether ignoring the correlation will increase or decrease the information (and therefore the covariance of position estimators) depends on both, the autocorrelation function and the geometrical configuration of the problem.

Although the behaviour of F_{xx} depends on the value of α, we can observe a characteristic pattern. First, the curve for the IID model ($\alpha = 1$) reaches the maximum always when beacons two and three coincide ($\phi = 0$), which implies a continuous increase in the amount of information as $\phi \to 0$. On the other hand, for $\alpha < 1$ the maximum is reached with $\phi > 0$. This means that, as beacons two and three move closer to one another, using models that account for the autocorrelation will eventually lead to a relative loss of information (negative slope as $\phi \to 0$). Figure 6.4e shows the value of F_{xx} for the limit case when $\phi = 0$ (beacons two and three overlapping) as a function of l_0 again for different values of α. Clearly, with this particular limit configuration, what model has the highest amount of information depends on l_0 and α.

Finally, it is interesting to note that when $\alpha = 0$ (purely exponentially decaying autocorrelation model), F_{xx} reaches the following limit value:

$$\lim_{\substack{\phi \to 0 \\ \alpha \to 0}} F_{xx} = \left(\frac{10n}{d\sigma \ln 10} \right)^2 \frac{2}{1-a}. \tag{6.31}$$

This expression is the same as the FIM of (6.29) (with $\alpha = 0$) calculated for the scenario with two beacons depicted in Figure 6.3; its value is plotted with a horizontal dashed line in figures 6.4b to 6.4e. This shows that, when using a purely exponentially decaying correlation model ($\alpha = 0$), having two beacons in overlapping positions provides the same information as having only one.

All in all, from the previous analysis we conclude that whether ignoring the autocorrelation results in more or less precise position estimators depends ultimately on the shape of the autocorrelation function and on the geometry of the problem, including the spatial distribution and density of the beacon placement, and that we cannot make general claims for arbitrary configurations. However, as we will see later, our real-life experiments show mostly a d/l_0 ratio that is better represented by figures 6.2b and 6.4b with $l_0 = 5$, in which the IID model clearly contains the largest amount of information. Therefore, we expect that taking the autocorrelation into account in the model will reduce the position estimator overconfidence.

6.4 Model Identification

In this section, we aim first to show experimental support for our choice of autocorrelation model, for which we will plot examples of the empirical correlograms calculated using our experimental data. Then we explain the details of the procedure used to estimate the model parameters.

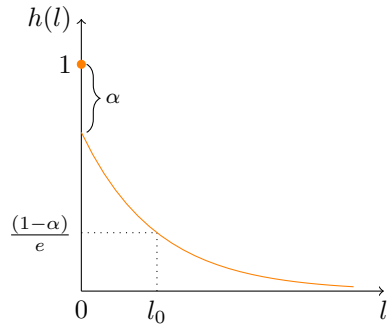


Figure 6.5: Theoretical autocorrelation function.

6.4.1 Spatial Correlogram

The autocovariance function, or covariogram, is defined as the covariance between observations in two positions. Our choice of autocovariance function is (6.16), which depends only on the distance between control points l. This distance is usually called *lag* among geo-statisticians, and we shall borrow their terminology here. Its corresponding autocorrelation function, or correlogram, is then

$$h(l) = \frac{k(l)}{k(0)} = \frac{\sigma_n^2 1_0(l) + \sigma_c^2 e^{-l/l_0}}{\sigma^2} = \alpha 1_0(l) + (1-\alpha)e^{-l/l_0}, \tag{6.32}$$

where recall that $\sigma^2 = \sigma_n^2 + \sigma_l^2$ and $\alpha = \sigma_n^2/\sigma^2$. Figure 6.5 presents its shape and main characteristic points. The local discontinuity in the origin has magnitude α. This gap is referred to as the *nugget* by geo-statisticians, and has a rather intuitive interpretation: the larger the relative dominance of the measurement noise over the shadowing (higher values of α) the wider the gap will be.

As we have seen, the autocorrelation function is related to the covariogram, and therefore one can estimate one from the other. Both are continuous functions of l (except in the origin in our case) that require the continuum of random variables $\boldsymbol{\varepsilon}$ for their calculation. In practice, however, one only has samples of the residuals with which they have to be estimated. Intuitively, in order to estimate e.g. the covariogram from residual samples, one could consider proceeding as follows: a) choose a lag l_{k}, b) select the data pairs (the residuals) from the training set whose lag is approximately l_{k}, and c) calculate the covariance between the data-pairs using the classical sample estimator of the covariance. Using this procedure for a set of selected lags, one can plot the covariance versus the lag for the points calculated, and then fit a function to these points.

In practice, geo-statisticians follow a similar procedure, but with some key differences. First, although the classical sample estimator of the covariance is

unbiased provided that the samples are independent, in our case the samples are not independent, and it can be shown that the previous estimator is biased with a bias that depends on the spatial dependences of the residuals, which is what we are trying to estimate. Therefore, the bias is difficult to correct and the sample covariance estimator is not generally used [Schabenberger and Gotway, 2004, p. 149],[Smith, 2014, chap. II. 4].

As an alternative, in practice geo-statisticians use the semi-variogram, which is defined as $\gamma(\varepsilon_{\mathrm{i}}, \varepsilon_{\mathrm{j}}) = \frac{1}{2}\mathrm{Var}\left[\varepsilon_{\mathrm{i}} - \varepsilon_{\mathrm{j}}\right]$, where for simplicity we denote $\varepsilon_{\mathrm{i}} = \varepsilon(\vec{\boldsymbol{r}}_{\mathrm{i}})$. For our second-order stationary spatial process, the semi-variogram can be expressed in terms of the covariogram as $\gamma(h) = k(0) - k(l) = \sigma^2 - k(l)$, and therefore one can be estimated from the other. Using the semi-variogram has several advantages, the most important being that its classical estimator, the empirical semi-variogram,[12] is unbiased. The usual procedure followed by geo-statisticians is then to first plot the empirical semi-variogram, then select the model upon visual inspection of the curve, and finally, fit it to the empirical semi-variogram, with which the model parameters are estimated [Schabenberger and Gotway, 2004, ch. 4].

In this thesis we will instead use ML to estimate the model parameters. Our goal in this subsection is simply present to the eye how the experimental correlation looks like, and then evaluate if our model is a reasonable one. Towards this end, we decided to present experimental correlograms from our data instead of the empirical semi-variograms, mainly for two reasons: first, the correlogram is dimensionless, and thus has a more intuitive interpretation. And second, with the large amount of data that we have available (about 60 datasets per environment), after experimenting with the semi-variogram and the correlogram we realized that the latter one is more adequate for our display purposes.

For a zero mean process, the experimental correlogram is:

$$\hat{k}(l) = \frac{1}{\hat{\sigma}|S_l|}\sum_{S_l}\varepsilon_i\varepsilon_j,$$

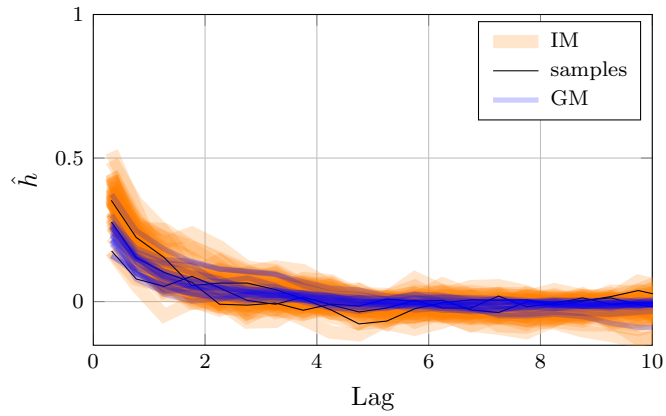
where

$$\hat{\sigma} = \frac{1}{N}\sum_{i=1}^{N}\varepsilon_i^2$$
$$S_l = \{(\varepsilon_i, \varepsilon_j) : ||\vec{\boldsymbol{r}}_{\mathrm{i}} - \vec{\boldsymbol{r}}_{\mathrm{j}}|| = l\}.$$

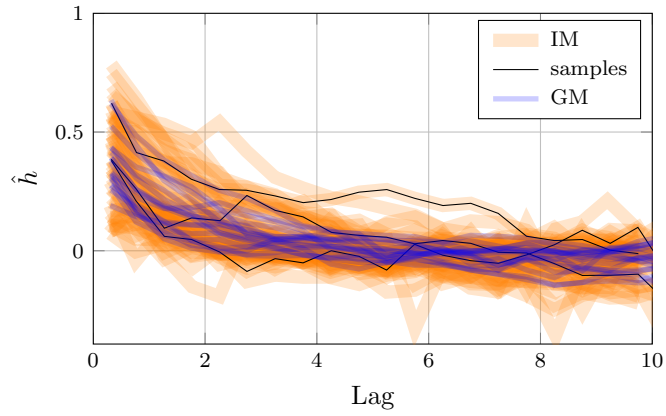
The set S_l is composed of point pairs whose distance is exactly l, and $|S_l|$ is its cardinality. In practice we rarely have control point pairs that are exactly at the same distance.[13] The empirical correlogram is then calculated for lag

[12] Proposed by Matheron in [Matheron, 1963]

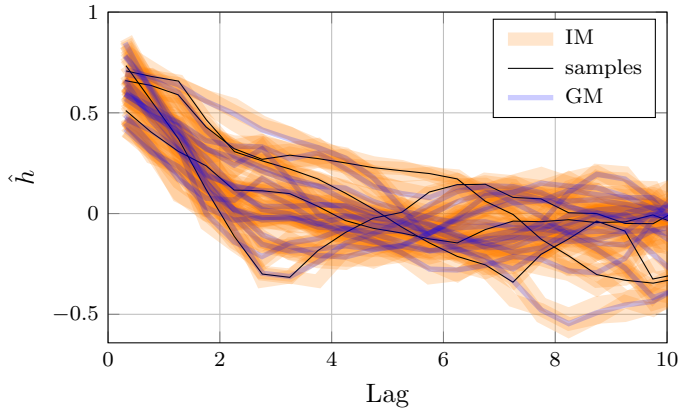
[13] This would be the case, for example, if we had the control points arranged in a regular



(a) Basketball field.



(b) Lobby



(c) Office

Figure 6.6: Experimental correlograms for the three environments

intervals rather than for a continuum of lags. For each lag interval, the average correlation is calculated using control point pairs whose distance is within the lag interval. Usually, these intervals are chosen to be contiguous and of equal size Δ. Considering P intervals, our estimator of the correlogram is then

$$\hat{k}(\bar{l}_p) = \frac{1}{\hat{\sigma}|S_p|} \sum_{S_p} \varepsilon_i \varepsilon_j \tag{6.33}$$

where

$$\begin{aligned} S_p &= \{(\varepsilon_i, \varepsilon_j) : ||\vec{\boldsymbol{r}}_{\mathrm{i}} - \vec{\boldsymbol{r}}_{\mathrm{j}}|| \in (l_p - \Delta/2, l_p + \Delta/2]\} \\ l_p &= p\Delta + \frac{\Delta}{2} \\ p &= \{0, \ldots, P-1\} \\ \bar{l}_p &= \frac{1}{|S_p|} \sum_{S_p} ||\vec{\boldsymbol{r}}_{\mathrm{i}} - \vec{\boldsymbol{r}}_{\mathrm{j}}||. \end{aligned}$$

For the previous estimator to be meaningful, for each lag interval there should be a significant number of samples $|S_p|$ with which we can calculate the averages. Usually this is taken as $|S_p| \geq 30$ [Smith, 2014]. Additionally, there should be a representative range of distances l_p for which we can calculate the averages using a sufficiently large number of samples. Ideally Δ should also be as small as possible, so that we will have larger number of intervals. But this decreases the number of measurements with which the averages are calculated in each interval. This normally requires a trade-off. In our case we have enough data so as to guarantee that, for a meaningful range of l, we have more than 30 measurements per lag interval with $\Delta = 0.5$ m.

Figure 6.6 shows the empirical correlograms calculated using (6.33) for all the datasets available within the different environments. The orange semi-transparent wide lines represent empirical correlograms calculated with data from a single node and dataset. Therefore, there are approximately 60 orange correlograms in each environment. The black opaque lines are examples of some of these curves, marked more clearly to stand out. The superposition of translucent orange lines creates darker areas that denote similarities between the different curves; or, from a different perspective, it emphasizes points where it will be more likely to find a curve representing the aggregated data. Finally, the semi-transparent blue lines are calculated grouping all the data gathered by each node in the 3 experiments done in each environment.

Panel 6.6a shows a clear similarity between the experimental correlogram and our model in the basketball field, and a larger relative dominance of the meas-

lattice.

urement noise with respect to the spatial shadowing ($\alpha < 0.5$). In the lobby (see Panel 6.6b), the concordance between the theoretical and empirical model is still visible. There is, however, a larger spreading of the curves. There are also cases in which, when using data from a single node and dataset, there is a significant deviation. In general, we can see an increase of the dominance in the shadowing with respect to the basketball field. In the office (see Panel 6.6c), the importance of the shadowing is clearly the largest among all the environments. Many curves show oscillations, and the validity of the model begins to be highly questionable in some cases. Nevertheless, the general aggregated trend still resembles a decaying exponential. For mathematical convenience and for the sake of simplicity, we then decide to use the same type of autocorrelation function in all environments.

6.4.2 Model Identification

Recall the model (6.17). In the model identification process we aim to find the value of $\boldsymbol{\theta}$ that results in an optimal fit of the model to a vector of observations $\boldsymbol{z} = (z_1, z_2, ..., z_{\mathrm{K}})^t$, where we defined $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathrm{m}}^t, \boldsymbol{\theta}_{\mathrm{c}}^t)^t = (-10n, z_0, \sigma_n, \sigma_c, l_0)^t$. The Gaussian process $\boldsymbol{\varepsilon}$ has, by definition, zero mean. Therefore, it is necessary to remove the trend $\bar{\boldsymbol{z}}$ before fitting the covariance model to the residuals. In order to do so, we first estimate $\hat{\boldsymbol{\theta}}_{\mathrm{m}} = (-10\hat{n}, \hat{z}_0)^t$ using least squares, resulting in

$$\hat{\boldsymbol{\theta}}_{\mathrm{m}} = (\mathbf{A}^t\mathbf{A})^{-1}\mathbf{A}^t\,\boldsymbol{z}, \tag{6.34}$$

where $\boldsymbol{A}$ is defined in (6.18). Using $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$, we then calculate the residuals,

$$\boldsymbol{\xi} = \boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}}, \tag{6.35}$$

whose likelihood and log-likelihood are, respectively

$$p(\boldsymbol{\xi}; \boldsymbol{\theta}_{\mathrm{c}}) = \frac{1}{\sqrt{(2\pi)^K |\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}\boldsymbol{\xi}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}\right\} \tag{6.36}$$

$$L(\boldsymbol{\xi}; \boldsymbol{\theta}_{\mathrm{c}}) = -\frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{\xi}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}) - \frac{K}{2}\ln(2\pi). \tag{6.37}$$

To find an estimate $\hat{\boldsymbol{\theta}}_{\mathrm{c}} = (\hat{\sigma}_l, \hat{\sigma}_c, \hat{l}_0)^t$ we maximize the log-likelihood of the residuals with respect to the three parameters, that is:

$$\hat{\boldsymbol{\theta}}_{\mathrm{c}} = \underset{\sigma_n, \sigma_c, l_0}{\operatorname{argmax}}\, L(\boldsymbol{\xi}; \boldsymbol{\theta}_{\mathrm{c}}).$$

As usual, the problem is non-linear and requires an iterative solver to find the maximum. We used the L-BFGS-B method from [Byrd et al., 1995], using the implementation available from [Zhu et al., 2011] and the Matlab interface from [Carbonetto, 2014], which had to be modified slightly to make it work in Octave.

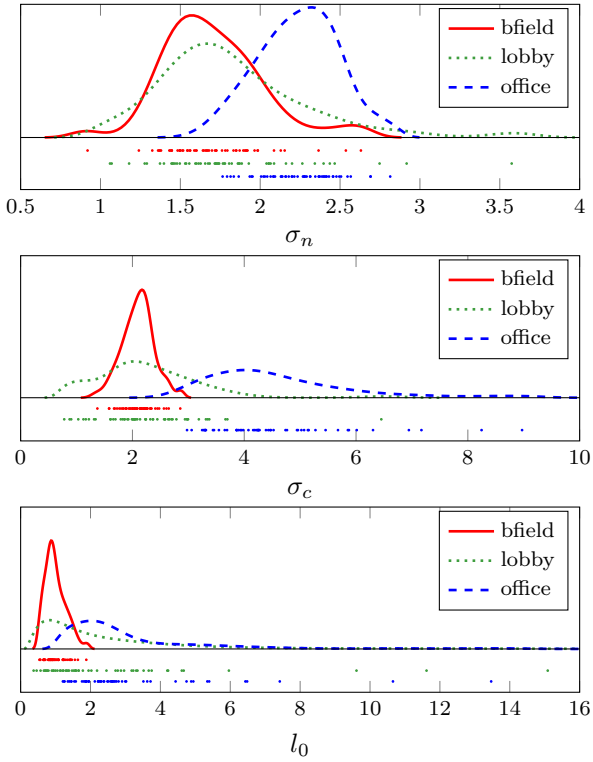


Figure 6.7: Distribution of σ_n, σ_c and l in the three environments (grouped datasets). The dots indicate the values used to estimate the distributions.

Table 6.1: Statistics of the distributions from Figure 6.7.

		Bfield	**Lobby**	**Office**
σ_n	mean	1.70	1.82	2.24
	median	1.66	1.74	2.25
	std	0.31	0.45	0.24
σ_c	mean	2.10	2.28	4.62
	median	2.14	2.14	4.25
	std	0.28	0.92	1.28
l_0	mean	0.99	3.09	3.23
	median	0.92	1.66	2.49
	std	0.29	4.64	2.33
σ	mean	2.72	3.00	5.17
	median	2.72	2.77	4.90
	std	0.26	0.74	1.15
α	mean	0.40	0.42	0.21
	median	0.38	0.37	0.23
	std	0.11	0.21	0.09

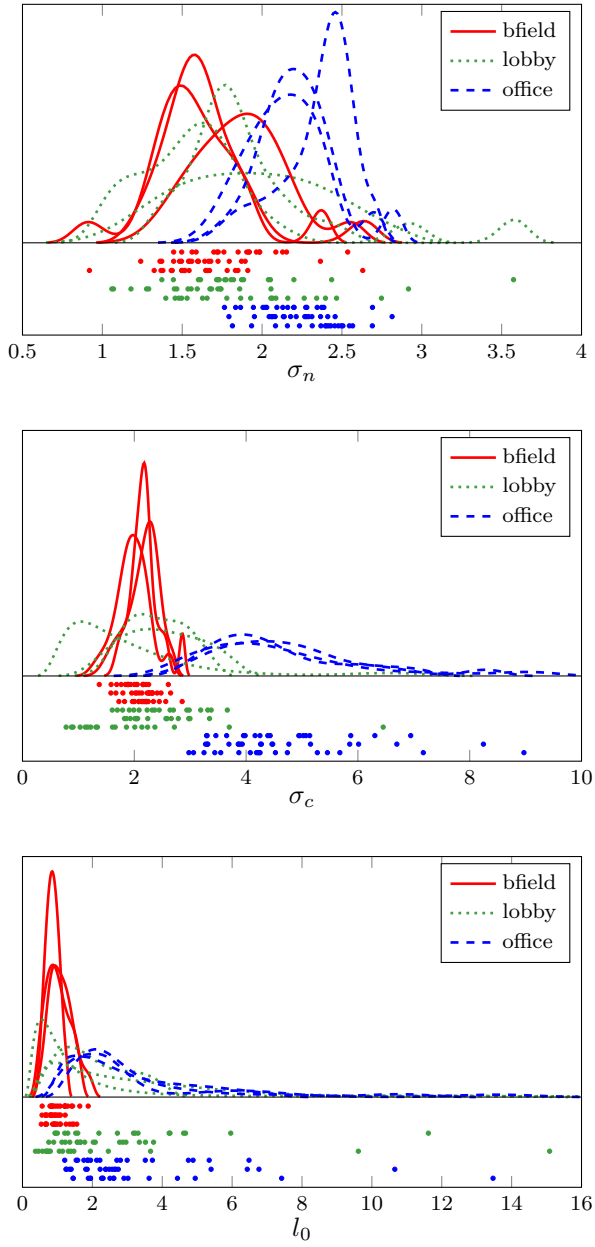


Figure 6.8: Distribution of σ_n, σ_c and l_0 in the three environments for the three robot trajectories. The dots indicate the values used to estimate the distributions.

Table 6.2: Mean, median and standard deviation of the distributions from Figure 6.8.

(a) Trajectory 1

		Bfield	**Lobby**	**Office**
σ_n	mean	1.86	1.91	2.14
	median	1.90	1.79	2.14
	std	0.26	0.49	0.23
σ_c	mean	1.97	2.51	4.62
	median	1.94	2.41	4.41
	std	0.27	0.62	1.02
l_0	mean	1.11	3.24	2.96
	median	1.08	2.95	2.46
	std	0.33	2.59	1.52

(b) Trajectory 2

		Bfield	**Lobby**	**Office**
σ_n	mean	1.63	1.60	2.21
	median	1.60	1.58	2.20
	std	0.25	0.43	0.22
σ_c	mean	2.16	2.42	4.63
	median	2.25	2.40	4.27
	std	0.27	0.47	1.32
l_0	mean	0.86	2.05	3.24
	median	0.85	1.72	2.44
	std	0.17	0.93	2.49

(c) Trajectory 3

		Bfield	**Lobby**	**Office**
σ_n	mean	1.60	1.94	2.35
	median	1.52	1.93	2.41
	std	0.34	0.40	0.23
σ_c	mean	2.18	1.92	4.61
	median	2.17	1.61	4.22
	std	0.26	1.32	1.51
l_0	mean	0.99	3.94	3.50
	median	0.92	1.15	2.56
	std	0.30	7.46	2.88

Similarly as we did in Chapter 3, we identify IMs from our experimental data. The initial parameters for the optimizer are calculated for each IM as follows. First we calculate the sample variance of the residuals $\widehat{\sigma^2}$, which we then divide equally between the measurement noise and the shadowing. That is, we use $\hat{\sigma}_n = \hat{\sigma}_c = (\widehat{\sigma^2}/2)^{1/2}$. For the decorrelation distance we select $l_0 = e^{\frac{\hat{\sigma}\ln 10}{10\hat{n}}} = 10^{\frac{\hat{\sigma}}{10\hat{n}}}$, using the recently estimated $\widehat{\sigma^2}$ and $\hat{n}$ from $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ in (6.34). The underlying idea is to choose a characteristic distance of the environment that grows according to its inhomogeneity, which is the main cause of the autocorrelation. We already found such a length when we were studying the different RSS-based distance estimators in Subsection 4.2.2, but in log-scale. Effectively, γ from (4.9) has dimension of (natural) log-distance. It appears in the CRLB of the distance and position estimators ((4.8) and (4.45) respectively), and in general in the statistics of the position estimators that we have developed in this thesis. It grows with σ, but pondered by the path-loss exponent n. Broadly speaking, n reflects more the density of obstacles, whereas σ the inhomogeneity. As the reader can verify, our choice for the initial decorrelation distance is $l_0 = e^{\hat{\gamma}}$, where $\hat{\gamma} = \frac{\hat{\sigma}\ln 10}{10\hat{n}}$ using the recently estimated $\widehat{\sigma^2}$ and $\hat{n}$.

Figure 6.7 presents the parameter values and distribution estimates of σ_n, σ_c and l_0 for each environment. Each PDF was estimated using the $3N$ model parameters available in each scenario[14] (represented as dots) and Gaussian kernel density estimation with positive support.[15] Table 6.1 presents the statistics of the distributions in both parameterizations, $(\sigma_n, \sigma_c, l_0)$ and (σ, α, l_0), where recall that $\sigma^2 = \sigma_n^2 + \sigma_c^2$ and $\alpha = \sigma_n^2/\sigma^2$. Figure 6.8 shows the densities and the parameter values for the data collected from each of the robot trajectories separately. Parameter values (dots) corresponding to the same robot trajectory are presented along the same horizontal line. The values of n and z_0 are the same as when ignoring the autocorrelation (see Figure 3.3). Table 6.2 presents the statistics for the distributions of Figure 6.8.

Figure 6.7 reveals the same general tendency of the parameter distributions as we observed in Chapter 3: their average value and dispersion increases with the clutterness and inhomogeneity of the environment. The narrowest distributions correspond to the basketball field, which is also the scenario in which the autocorrelation model is more adequate. Table 6.1 shows the clear increasing tendency of the total variance σ^2, together with a significant gain in the relative importance of the shadowing in the office with respect to the measurement noise (smaller value of α in the office). This shows concordance with the experimental correlograms of Figure 6.6, where we saw that the highest values closest to the origin are attained in the office. The dispersion of l_0 is quite remarkable in both

[14] One for each of the N nodes and three robot trajectories

[15] Code available at `http://www.mathworks.co.uk/matlabcentral/fileexchange/19121`

the lobby and the basketball field.[16] We believe that this large dispersion is partly a consequence of the model not being explanatory enough. Figure 6.8 exposes the potential variability of the densities depending on the particular dataset used, even within the same environment. This inter-experiment variability is more accentuated in the lobby, where the presence of large dominant obstacles makes every experiment quite unique.

6.5 ML Localization Discarding Samples

One way to deal with the autocorrelation is simply to use only uncorrelated measurements, with which we can then use the IID assumption and the localization methods studied in Chapter 5. This section is dedicated to exploring this approach using ML localization.

In order to select uncorrelated samples from a dataset, we need a mechanism for discarding correlated observations. Intuitively, measurements taken further apart will less likely be similar than others taken in closer positions. Or in the words of Tobler and his first law of geography, '*Everything is correlated with everything else, but close things are more correlated than things that are far away*' [Tobler, 1970]. The essence of this law is captured in our model, in which the autocorrelation decreases exponentially with the distance between control points. After a long enough distance we can assume that the correlation is negligible. Our strategy here is then to select only the measurements corresponding to control points that are separated from each other at least a distance l_d.

The problem of selecting the control points can be formulated mathematically as follows: given a set of K points in space $S = \{\vec{\boldsymbol{r}}_{\mathrm{k}}, k = 1 : K\}$, find a subset $S_s \subseteq S$ so that $||\vec{\boldsymbol{r}}_{\mathrm{i}} - \vec{\boldsymbol{r}}_{\mathrm{j}}|| > l_d \; \forall \; \vec{\boldsymbol{r}}_{\mathrm{i}}, \vec{\boldsymbol{r}}_{\mathrm{j}} \in S_s$. There might be many possible subsets that satisfy the previous condition. Ideally we want to discard the minimum possible number of measurements, that is to say: we want to find the largest possible subset S_s. This leads to a non-trivial combinatorial optimization problem. Instead of aiming for an optimal solution, we propose[17] an intuitive practical heuristic with reproducible results which gives a reasonably good output. Table 6.3 shows a simplified implementation in Octave/Matlab.

The algorithm accepts as inputs the $K \times 2$ vector of control points, `b_pos` (the position of the beacons), the minimum distance among the control points, `l_d`, and the index of an element of `b_pos` with which the algorithm will start,

[16] We shall mention here that there was an outlier in the lobby with $l_0 = 30.68$ m observed from the data obtained in the third robot trajectory, which we decided to leave out of the figures for the sake of clarity. This explains the large difference between the mean and the median values of l_0 in the lobby in tables 6.2c and 6.1.

[17] Thanks to Andrè Schumacher, who initially proposed this algorithm to the author of this thesis.

Table 6.3: Algorithm to discard correlated samples.

```
function b_pos_sel=distant_subset(b_pos,l_d,seed_i)
  rem_i=1:rows(b_pos);
  sel_i=seed_i;
  D=distmat(b_pos);
  r_i=find(D(:,seed_i)<l_d);
  D(:,r_i)=[];D(r_i,:)=[];
  rem_i(r_i)=[];
  while length(rem_i)>0
    [s,idx]=sort(sum(D<l_d)-1);
    s_i=idx(1);
    sel_i=[sel_i rem_i(s_i)];
    r_i=find(D(:,s_i)<l_d);
    D(:,r_i)=[];D(r_i,:)=[];
    rem_i(r_i)=[];
  endwhile
  b_pos_sel=b_pos(sel_i,:);
```

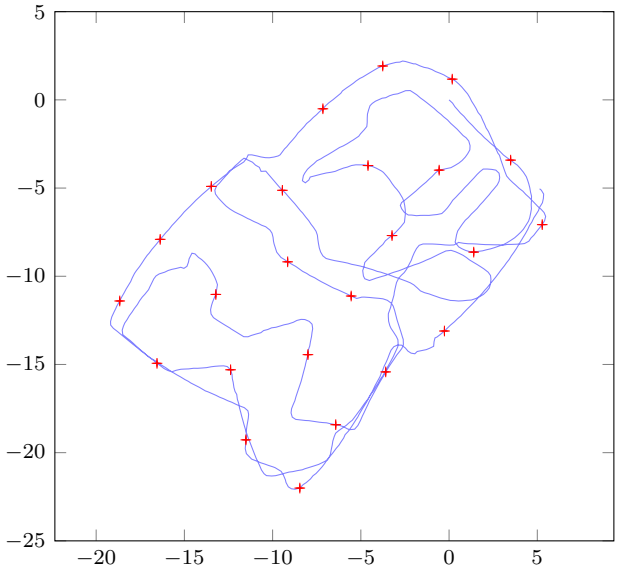


Figure 6.9: Example of execution of the sample selection algorithm from Table 6.3 in the basketball field ($l_d = 4$). The '+' signs represent the selected points.

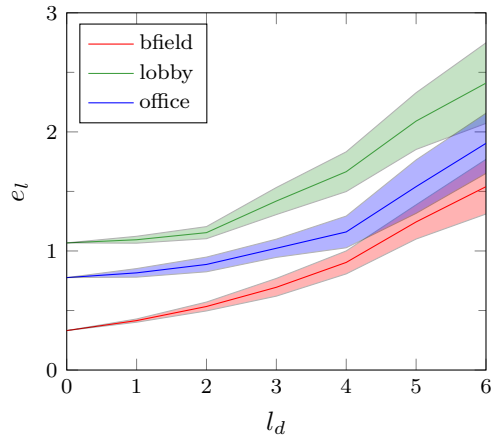
`seed_i`. Lines 2 and 3 create vectors to store the indices of the remaining and selected control points, respectively. Initially `rem_i` contains all the elements, while `sel_i` contains only the seed. Line 4 computes the KxK distance matrix among elements of `b_pos`, that is: `D(i,j)` contains the distance between the i-th and j-th control points. Line 5 finds the indices of the control points that are closer than a distance `l_d` to the seed `seed_i`. These indices are removed from the reminder pool and the matrix `D` in lines 6 and 7. Therefore, all the points in `rem_i` are further than `l_d` to `seed_i`. Then a while loop starts, which will continue until the pool of reminding control points is not empty.

Lines 9 and 10 find the control point among the reminders with the least number of neighbours distant less than `l_d` to all of the control points already selected. This point is then added to the group of selected control points in line 11. Line 12 finds the control points among the reminders closer than `l_d` to the newly selected point, and lines 13 and 14 remove them from the list of reminders and `D`. Upon completion, the indices in `sel_i` correspond to control points at a distance of at least `l_d` from each other. The results from an sample execution of this algorithm in the basketball field for $l_d = 4$ can be seen in Figure 6.9.

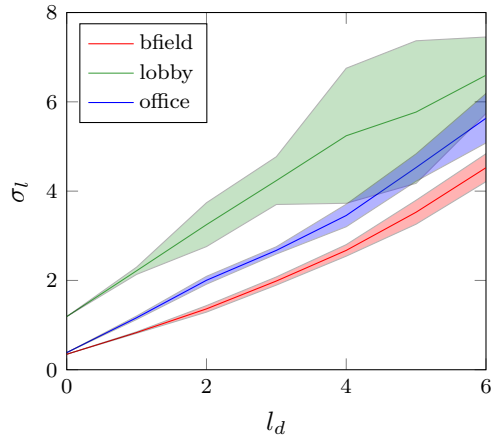
The panels from Figure 6.10 show the effects of discarding samples in ML localization using the previous algorithm as a function of l_d in our three different experimental environments. They were created calculating the average of the localization error and its standard deviation[18] for IMs as described in Subsection 5.3.1, using only the measurements selected by the algorithm in Table 6.3. We denote this (average) error and (average) standard deviation as e_d and σ_d, respectively. Note that, due to the random nature of the sample selection algorithm, e_d and σ_d are random variables themselves. For each l_d we calculate 20 realizations of the pair (e_d, σ_d) using a random seed in the sample selection algorithm. Figure 6.10 shows the value of e_d, σ_d and the ratio e_d/σ_d as a function of l_d. The solid lines represent average values, and the shaded areas the region around the average plus/minus two standard deviations.

Regarding the selection of l_d, as explained before we aim at choosing uncorrelated samples, and in that sense l_d should be as large as possible. On the other hand, as we discard samples we are also discarding information, which can result in an increase of the localization error. Panel 6.10a confirms the tendency of the mean error to increase together with l_d. While discarding samples increases the average error e_l, the standard deviation σ_l also increases naturally (see Panel 6.10b), but at a different rate. Panel 6.10c shows that, for the range of l_d represented, σ_l increases faster than e_l, resulting in a decreasing e_l/σ_l ratio. The implication of this is a relatively less biased position estimator as l_d increases, meaning that the true position is more likely to be encountered within the error ellipse for the same confidence level.

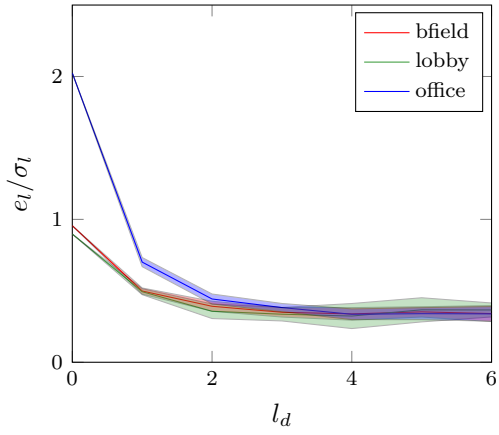
[18] The square root of the trace of the covariance matrix



(a) Average ML localization error.



(b) Average standard deviation of ML localization error.



(c) e_l/σ_l ratio.

Figure 6.10: Effects of discarding samples vs. minimum distance among samples.

The technique of discarding samples leaves to the designer the selection of l_d. If the type of environment is known, one option is to simply adopt values obtained from prior studies. If training data is available, one can plot the autocorrelation function and decide upon. When using exponentially decaying models for the autocorrelation, a natural choice is to select the decorrelation distance $l_d = d_0$. Using the medians found in Table 6.1 as the reference values for l_0, from the data used to plot Panel 6.10a we obtain that the increments of the average errors are of about 10, 6 and 18 cm in the basketball field, lobby and office respectively, which correspond to increases of 27%, 5% and 18% with respect to using full datasets. The resulting data size reductions are of more than 80% in all cases.

Based on the studies and discussion provided throughout this section, we conclude that discarding measurements can solve the overconfidence problem related to the IID errors assumption, but at the price of a larger average error. The advantages are the relative decrease of the bias and the saving in number of computations, as the reduction of the dataset can be significant. A key drawback is the need to manually select the minimum distance among samples l_d, which in the absence of proper training data would be a subjective decision hopefully made by an educated eye. In the following section, instead of discarding samples we use the full dataset and let the autocorrelation model tackle the overconfidence problem. This eliminates the need for deciding l_d, although at the expenses of requiring a properly identified model. In any case, incorporating the solution within the model opens the door to approaches that jointly estimate the position and model parameters, methods that we will study in the next chapter.

6.6 ML Localization Using Full Data

In this section, our goal is to present the localization results that validate in practice the hypothesis that incorporating autocorrelation into the model solves the over-confidence problem, up to the point allowed by the ever-present log-normal model misspecification.

Under the log-normal model with autocorrelation (6.17), the likelihood of a vector of K measurements $\boldsymbol{z} = (z_1, z_2, ..., z_K)^t$ is

$$p(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}) = \frac{1}{\sqrt{(2\pi)^K |\boldsymbol{\Sigma}|}} \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \bar{\boldsymbol{z}})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{z} - \bar{\boldsymbol{z}})\right\}, \tag{6.38}$$

and the log-likelihood

$$L(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}) = -\frac{1}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2}(\boldsymbol{z} - \bar{\boldsymbol{z}})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{z} - \bar{\boldsymbol{z}}) - \frac{K}{2} \ln (2\pi). \tag{6.39}$$

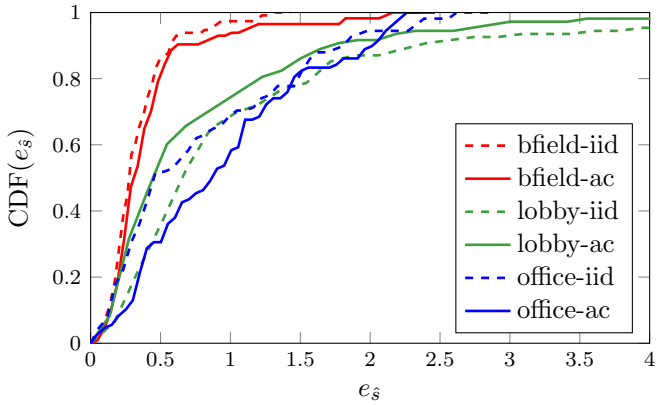


Figure 6.11: CDF of the error.

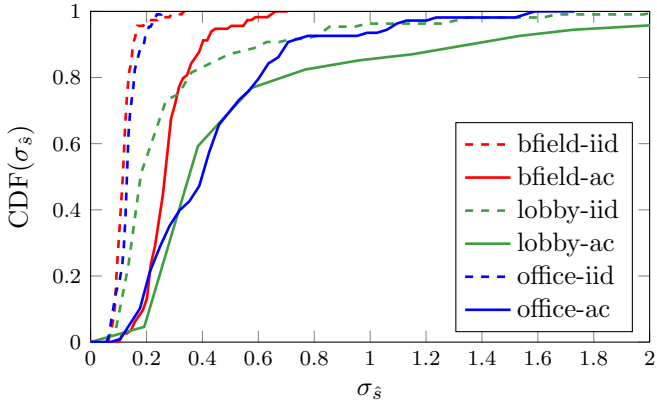
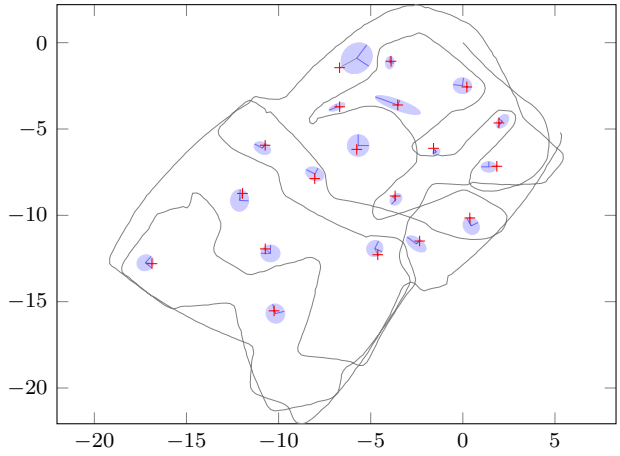


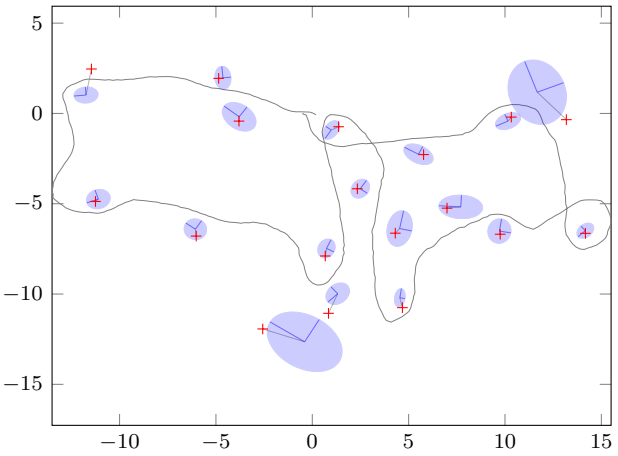
Figure 6.12: CDF of the estimator's standard deviation.

Table 6.4: Comparison between ML position estimates when ignoring (IID) and modelling (AC) autocorrelation (distance in metres).

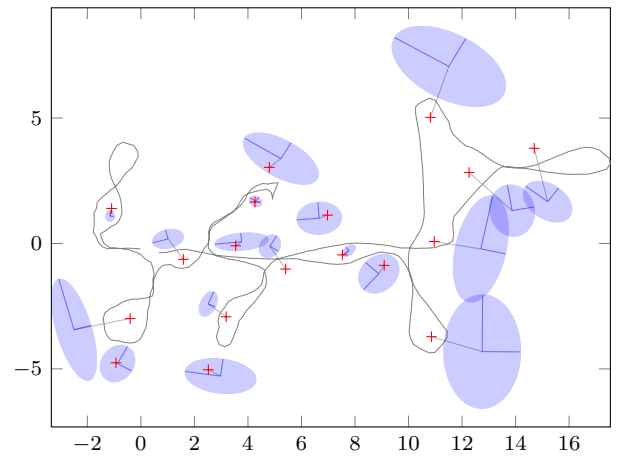
	IID			**AC**		
	$\bar{e}_{\hat{s}}$	$\bar{\sigma}_{\hat{s}}$	$\bar{e}_{\hat{s}}/\bar{\sigma}_{\hat{s}}$	$\bar{e}_{\hat{s}}$	$\bar{\sigma}_{\hat{s}}$	$\bar{e}_{\hat{s}}/\bar{\sigma}_{\hat{s}}$
Basketball field	0.33	0.12	2.78	0.41	0.28	1.45
Lobby	1.07	0.30	3.62	0.79	0.64	1.25
Office	0.78	0.13	5.84	0.94	0.44	2.16



(a) Basketball field.



(b) Lobby



(c) Office

Figure 6.13: Example of ML localization results when considering exponentially decaying autocorrelation.

Using the values of $\boldsymbol{\theta}$ estimated during the model identification, the ML position estimator is

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{ML}} = \underset{\vec{\boldsymbol{s}}}{\operatorname{argmax}}\, L(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}).$$

This, once more, is a non-linear problem that we solve using the L-BFGS-B solver initialized with the results from a grid-search with a resolution of 0.5 m. The main differences with respect to the log-likelihood under the IID assumption (see (5.1)) is the presence of the matrix $\boldsymbol{\Sigma}^{-1}$ weighting the residuals, which can effectively change the ML position estimate. Because $\boldsymbol{\Sigma}$ does not depend on the variable $\vec{\boldsymbol{s}}$, its inverse needs to be calculated only once,[19] and it can be stored for all the necessary evaluations of the log-likelihood. Once the solver has converged, the covariance of the position estimate is calculated as

$$\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{ML}}\right] = -\boldsymbol{H}|^{-1}_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{ML}}},$$

where $\boldsymbol{H}|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{ML}}}$ is the Hessian of the log-likelihood function with respect to the position $\vec{\boldsymbol{s}}$ evaluated in the maximum $\hat{\vec{\boldsymbol{s}}}_{\mathrm{ML}}$ (see Section 4.6).

Table 6.4, Figure 6.11 and Figure 6.12 present a comparison of ML localization results in our three experimental environments when assuming IID versus autocorrelated measurements. As usual, we calculate position estimates using IMs and the cross-validation scheme described in Subsection 5.3.1. For each position estimate $\hat{\vec{\boldsymbol{s}}}^{\mathcal{S}}_{\mathrm{ijk}}$, in addition to calculating the error $e_{ijk} \equiv \left|\left|\vec{\boldsymbol{s}}_{\mathrm{i}} - \hat{\vec{\boldsymbol{s}}}^{\mathcal{S}}_{\mathrm{ijk}}\right|\right|$ we also calculate its corresponding variance $\sigma^2_{ijk} \equiv \mathrm{Tr}\left(\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}^{\mathcal{S}}_{\mathrm{ijk}}\right]\right)$. The outcomes e_{ijk} and σ_{ijk} are then realizations of random variables, which, for simplicity and for the remainder of this section only, we denote as $e_{\hat{s}}$ and $\sigma_{\hat{s}}$, respectively. Figures 6.11 and 6.12 present the empirical CDF of $e_{\hat{s}}$ and $\sigma_{\hat{s}}$. Results associated with the IID assumption are plotted as dashed lines and labelled with the -iid suffix. Results corresponding to our autocorrelation model are presented with solid lines and labelled with the -ac suffix. Table 6.4 presents the mean values of $e_{\hat{s}}$ and $\sigma_{\hat{s}}$.

From Table 6.4 we see that the mean localization error increases about 8 and 16 cm in the basketball field and office respectively when using the model with autocorrelation, which represent a 20% of growth. However, in the lobby it decreases 28 cm, a respectable 26% reduction. On the other hand, the standard deviation of the position estimators on average increases quite clearly and notably in all cases, increasing more than 130% in the basketball field and lobby and 230% in the office. When comparing the $\bar{e}_{\hat{s}}/\bar{\sigma}_{\hat{s}}$ ratio, we see that using the model with autocorrelation results in a significant decrease in all environments. Therefore, the estimators using models accounting for the autocorrelation are relatively less biased than those assuming IID errors.

[19] In reality the inverse is never calculated explicitly

Figure 6.13 shows examples of localization results from the three different environments. These shall be compared with those from Figure 5.5, which presents the results when ignoring the autocorrelation. The error ellipses correspond to a 95% confidence level in all cases. The increase of the covariance of position estimates when considering the autocorrelation is clear from the comparison. In the basketball field, all the position estimates have a small error in Panel 5.5a, but the true positions (red '+' signs) clearly fall outside of the error ellipses in most of the cases. This is the effect of overconfidence, which makes the error ellipses too small. Panel 6.13a shows that this issue is clearly corrected as a result of considering the autocorrelation in the model. We can see a similar effect in the lobby (panels 6.13b and 6.13b). Additionally, as expected, position estimates corresponding to nodes in the outer parts of the robot trajectories have a larger error and bias due to the worse geometrical configuration. In the office, assuming IID measurements results in clearly biased position estimates (see Panel 5.5c). When considering the autocorrelation (see Panel 6.13c), the bias is still very appreciable, which, as already argued, is mostly due to the model misspecification. However, the information about the *trust* that we have on our estimates is much more realistic than in Panel 5.5c.

All in all, the main effect of considering the autocorrelation in the model is not necessarily an increase in the precision and accuracy of position estimates, but rather an increase in their credibility and plausibility in the sense that they are relatively less biased and better reflect the uncertainty of the estimates. When compared with the strategies of information reduction and samples rejection studied in sections 5.1 and 6.5, respectively, including the autocorrelation in the model did not result in the smallest average error (see table 6.4). Moreover, in the basketball field and office, where the average error increases, we can see in Figure 6.10a that the same or better results would have been achieved by selecting only samples 1 m and 2.5 m apart, respectively, leading to a considerable reduction in the size of the dataset and a corresponding computational savings. Considering the autocorrelation in the model still holds two major advantages: a) one does not need to select manually any parameter, as is the case with the information reduction and samples rejection strategies, and b) the formulation of the problem allows a natural treatment of these parameters as nuisance quantities that can be e.g. marginalized out or estimated jointly together with the position and the rest of the model parameters, as we will do in the next chapter.

6.7 LLS Localization

In Chapter 4 we formulated the equations of three families of linear position estimators on the basis of the IID errors assumption. Here we provide the equivalent formulation considering autocorrelated errors under our choice of autocorrelation model.

Recall the linear system $\boldsymbol{A}_{\mathrm{L}}\vec{\boldsymbol{s}} = \boldsymbol{b}_{\mathrm{L}}$ from (4.84), where we substituted the true $\boldsymbol{b}_{\mathrm{L}}$ with $\hat{\boldsymbol{b}}_{\mathrm{L}}$ defined as in (4.92), which is formed using estimates of the squared distances. We studied three different families of position estimators based on three different types of solution: OLS, GLS and IGLS. The main formulas/procedures used by these estimators do not change when adding the spatial autocorrelation to the RSS model; only the values of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ and its covariance $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$. Therefore, the OLS and GLS position estimators are still given by (4.85) and (4.100), respectively, and the algorithm for the IGLS is the modified one presented in Table 5.5. What remains to be done is to re-calculate $\hat{\boldsymbol{b}}_{\mathrm{L}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ when using the three different distance estimators studied in this thesis: ML , unbiased and LMMSE. We now present the results of these calculations, leaving the derivation details for Appendix B.3.

Following the notational scheme followed in Subsection 4.2.3, let us call $\hat{P}_{\mathrm{i}}$ the estimate of d_i^m. The complete vector of estimates for the m-th power of the K distances is then $\hat{\boldsymbol{P}} = (\hat{P}_{1}, \ldots, \hat{P}_{\mathrm{K}})^t$. Let us further call $\breve{\boldsymbol{P}}_{\boldsymbol{u}} = (\hat{P}_{1}, \ldots, \hat{P}_{i-1})^t$ and $\breve{\boldsymbol{P}}_{\boldsymbol{l}} = (\hat{P}_{i+1}, \ldots, \hat{P}_{\mathrm{K}})^t$, so that $\hat{\boldsymbol{P}} = (\breve{\boldsymbol{P}}_{\boldsymbol{u}}^t, \hat{P}_{\mathrm{i}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}^t)^t$ is the complete vector of distances power estimates. That is:

$$\breve{\boldsymbol{P}}_{\boldsymbol{u}} \equiv \begin{pmatrix} \hat{P}_{1} \\ \vdots \\ \hat{P}_{\mathrm{i}-1} \end{pmatrix} \quad \text{and} \quad \breve{\boldsymbol{P}}_{\boldsymbol{l}} \equiv \begin{pmatrix} \hat{P}_{\mathrm{i}+1} \\ \vdots \\ \hat{P}_{\mathrm{K}} \end{pmatrix}, \quad \text{leading to} \quad \hat{\boldsymbol{P}} = \begin{pmatrix} \breve{\boldsymbol{P}}_{\boldsymbol{u}} \\ \cdots\cdots \\ \hat{P}_{\mathrm{i}} \\ \cdots\cdots \\ \breve{\boldsymbol{P}}_{\boldsymbol{l}} \end{pmatrix}.$$

The covariance $\mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right]$ can be calculated from $\mathrm{Cov}\left[\hat{\boldsymbol{P}}\right]$. When using the ML, unbiased and LMMSE distance estimators, respectively, this covariance takes the following values:

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{ML}}\right] = \boldsymbol{D}_{\boldsymbol{m}} \odot \boldsymbol{Q} \tag{6.40}$$

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{U}}\right] = \frac{1}{\Gamma_m^2}\boldsymbol{D}_{\boldsymbol{m}} \odot \boldsymbol{Q} \tag{6.41}$$

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{V}}\right] = \frac{1}{\Gamma_m^6}\boldsymbol{D}_{\boldsymbol{m}} \odot \boldsymbol{Q}, \tag{6.42}$$

where $\odot$ is the Hadamard or element-wise product between matrices and we use

the following definitions:

$$\boldsymbol{D_m} \equiv \begin{pmatrix} d_1^m d_1^m & d_1^m d_2^m & \cdots & d_1^m d_{\text{K}}^m \\ d_2^m d_1^m & d_2^m d_2^m & \cdots & d_2^m d_{\text{K}}^m \\ \vdots & \vdots & \ddots & \vdots \\ d_{\text{K}}^m d_1^m & d_{\text{K}}^m d_2^m & \cdots & d_{\text{K}}^m d_{\text{K}}^m \end{pmatrix} \tag{6.43}$$

$$\boldsymbol{Q} \equiv \Gamma_m^2 \left[\exp\left\{ \gamma_{mn}^2 \mathbf{I} + \gamma_{mc}^2 \boldsymbol{W} \right\} - \mathbf{1}\mathbf{1}^t \right] \tag{6.44}$$

$$\Gamma_m \equiv e^{\gamma_m^2/2} \tag{6.45}$$

$$\gamma_m^2 \equiv \gamma_{mn}^2 + \gamma_{mc}^2 \tag{6.46}$$

$$\gamma_{mn} \equiv \frac{m\sigma_n \ln 10}{10n}, \quad \gamma_{mc} \equiv \frac{m\sigma_c \ln 10}{10n}, \tag{6.47}$$

where the exponential in (6.44) is element-wise and $\boldsymbol{W}$ is defined in (6.19).

We now partition the covariance of $\hat{\boldsymbol{P}}$ in the following blocks:

$$\text{Cov}\left[\hat{\boldsymbol{P}}\right] = \left(\begin{array}{c:c:c} \beta_{uu} & \beta_{ui} & \beta_{ul} \\ \hdashline \beta_{iu} & \beta_{ii} & \beta_{il} \\ \hdashline \beta_{lu} & \beta_{li} & \beta_{ll} \end{array}\right) = \left(\begin{array}{c:c:c} \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] & \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \hat{P}_{\text{i}}\right] & \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}\right] \\ \hdashline \text{Cov}\left[\hat{P}_{\text{i}}, \breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] & \text{Cov}\left[\hat{P}_{\text{i}}, \hat{P}_{\text{i}}\right] & \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}\right] \\ \hdashline \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{l}}, \breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] & \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \hat{P}_{\text{i}}\right] & \text{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}\right] \end{array}\right). \tag{6.48}$$

Using the respective values of $\text{Cov}\left[\hat{\boldsymbol{P}}\right]$, the covariance $\text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right]$ is finally calculated as

$$\boldsymbol{\Sigma}_{\mathbf{L}} = \frac{1}{4}\left\{ \left(\begin{array}{c:c} \beta_{uu} & \beta_{ul} \\ \hdashline \beta_{lu} & \beta_{ll} \end{array}\right) + \beta_{ii}\mathbf{1}\mathbf{1}^t - \left[\begin{pmatrix} \beta_{ui} \\ \hdashline \beta_{li} \end{pmatrix} \mathbf{1}^t + \mathbf{1} \left(\begin{array}{c:c} \beta_{iu} & \beta_{il} \end{array} \right) \right] \right\}. \tag{6.49}$$

The reader can verify that, in the limiting cases of $\sigma_c \to 0$ or $l_{\text{o}} \to 0$ in which the correlation vanishes, these formulas for $\boldsymbol{\Sigma}_{\mathbf{L}}$ simplify to those calculated in Chapter 4 under the IID measurements assumption (see (4.94), (4.96) and (4.98)).

Note that the matrix of distance power products, $\boldsymbol{D_m}$, depends on the true node-beacon distances, which are unknown. Again, as we did in Chapter 4, in practice we can approximate this matrix using the estimated powers of distances,

leading to

$$\hat{\boldsymbol{D}}_{\boldsymbol{m}} = \begin{pmatrix} \widehat{d_1^m}\widehat{d_1^m} & \widehat{d_1^m}\widehat{d_2^m} & \cdots & \widehat{d_1^m}\widehat{d_{\mathrm{K}}^m} \\ \widehat{d_2^m}\widehat{d_1^m} & \widehat{d_2^m}\widehat{d_2^m} & \cdots & \widehat{d_2^m}\widehat{d_{\mathrm{K}}^m} \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{d_{\mathrm{K}}^m}\widehat{d_1^m} & \widehat{d_{\mathrm{K}}^m}\widehat{d_2^m} & \cdots & \widehat{d_{\mathrm{K}}^m}\widehat{d_{\mathrm{K}}^m} \end{pmatrix} = \hat{\boldsymbol{P}}\hat{\boldsymbol{P}}^t. \tag{6.50}$$

Replacing $\boldsymbol{D_m}$ for $\hat{\boldsymbol{D}}_{\boldsymbol{m}}$ in (6.40), (6.41) and (6.42) allows us then to calculate their respective estimates $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$.

In summary, the procedure for calculating a position estimate is as follows:

1. Select the distance estimator to be used and calculate $\hat{\boldsymbol{P}}$ as explained in Subsection 4.2.3 using the exponent $m = 2$ and the definitions of γ_m and Γ_m provided in 6.45 and 6.46, respectively.
2. Calculate $\hat{\boldsymbol{D}}_{\boldsymbol{m}}$ using (6.50) using the recently calculated $\hat{\boldsymbol{P}}$.
3. Calculate Cov $\left[\hat{\boldsymbol{P}}\right]$ using (6.40), (6.41) or (6.42) depending on the distance estimator chosen, in which $\hat{\boldsymbol{D}}_{\boldsymbol{m}}$ substitutes for $\boldsymbol{D_m}$.
4. Decompose Cov $\left[\hat{\boldsymbol{P}}\right]$ in blocks as in (6.48).
5. Calculate $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ using (6.49).
6. Calculate $\hat{\boldsymbol{b}}_{\mathrm{L}}$ from the values of $\hat{\boldsymbol{P}}$ using (4.92).

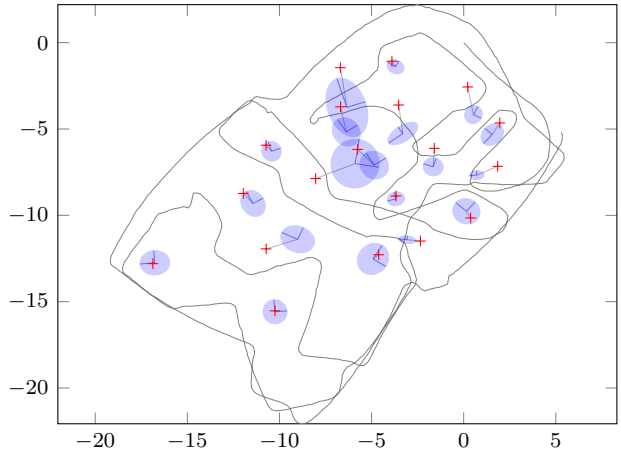
The values of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ and $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ are then used by the OLS, GLS and IGLS position estimators given by equation (4.85), equation (4.100) and Table 5.5, respectively.

The mean localization errors when using IMs, the previous algorithms and the data from our experiments are summarized in Table 6.5. These shall be compared with those from Table 5.4b, which present the results under the IID measurements assumption. The mean errors when using the OLS/ML position estimator are the same as those that result when the autocorrelation is not considered. This is as expected, because the ML distance estimator depends only on the path-loss exponent and reference power, which are the same for both models. Such is not the case when using the unbiased and LMMSE distance estimators; they both depend on Γ_m, which gets a different value depending on whether or not the model considers the autocorrelation. As seen in the tables, the errors for the OLS estimators when using the unbiased and LMMSE distance estimators are slightly different when ignoring and taking into account the correlation. However, the GLS and IGLS algorithms experience a significant general increase in the error in all cases when considering the correlation.

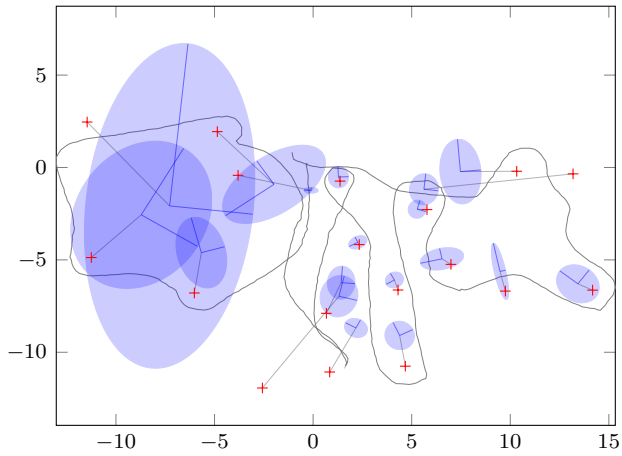
Table 6.5: Comparison of average errors of LLS position estimates in different environments using IMs and autocorrelated noise. Distance in metres

		ML	Unbiased	LMMSE
OLS	**Basketball field**	3.05	1.71	3.18
	Lobby	56.24	21.13	7.55
	Office	4.91	2.44	2.97
GLS	**Basketball field**	1.18	1.23	1.30
	Lobby	3.23	3.37	3.54
	Office	1.85	1.85	1.86
IGLS	**Basketball field**	1.03	1.18	1.17
	Lobby	1.91	2.20	2.98
	Office	1.95	2.16	1.60

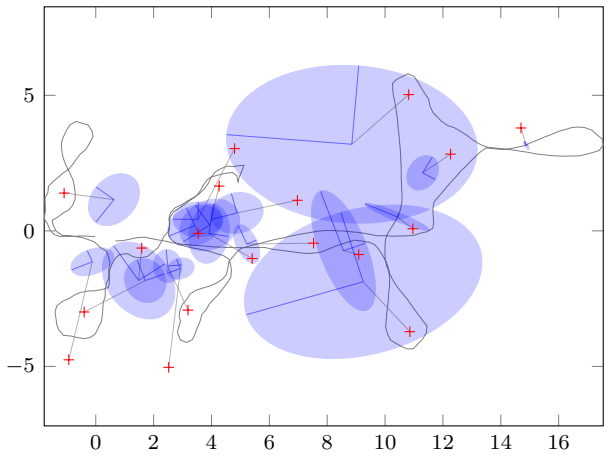
Figure 6.14 presents examples of localization results from the three experimental environments when using he IGLS algorithm with ML distance estimates, the algorithm that had the smallest average errors. These shall be compared with those from Figure 5.7, which were calculated under the IID errors assumption using the same data and algorithm. In the basketball field, we see a relative increase of the covariances when considering the autocorrelation. However, the average error increases significantly as well, making the estimates still relatively biased. In the lobby and the office the situation is similar, but characterized by a much larger increase of the covariance and bias. There is a clear tendency of the estimates to be biased towards the centre of the area englobed by the robot trajectory, perhaps a symptom of overly short distance estimates. We leave the investigation of the causes of these phenomena for later work. In any case, ignoring the autocorrelation clearly resulted in better position estimates when using our model and in our experiments.



(a) Basketball field.



(b) Lobby



(c) Office

Figure 6.14: Example of LLS localization results when considering exponentially decaying autocorrelation.

6.8 Conclusions

When using log-normal RSS-distance models, the residuals exhibit autocorrelation due to shadowing. This is a direct consequence of the model miss-specification, which results in a model incapable of explaining the non-radial relations that appear in inhomogeneous environments. As a consequence of this structural model limitation, the residuals become spatially correlated.

One way to tackle this issue is to incorporate the autocorrelation into the RSS model. To do this we chose a zero mean, homogeneous and isotropic Gaussian process with exponentially decaying autocorrelation function to model the shadowing, while retaining a white noise term to account for any other source of uncertainty. The empirical correlograms presented show a good adequacy of the model for the basketball field, slightly worse for the lobby and more questionable for the office, where oscillations were not infrequent. Nevertheless, the overall general trend in the correlograms is a decay, and for simplicity and mathematical convenience we still decided to use the same model in all environments.

Compared with considering IID errors, the main effect of incorporating the autocorrelation into the RSS model in ML localization is an increase of the estimator's covariance. This results in relatively less biased and more realistic position estimates, and it partially addresses the overconfidence problem observed in Chapter 5. However, position estimates in clearly inhomogeneous environments can still be biased due to the model misspecification.

Another possibility to deal with the autocorrelation is to discard correlated measurements and use localization methods that rely on the IID errors assumption. In the presented approach, this requires finding control points further apart from one another a minimum distance l_d, for which we have proposed a practical and simple heuristic. The results of our analysis shows that, although the covariance of the estimates increases effectively, the mean error can also increase with the number of discarded observations because less information is used.

The main advantage of discarding samples is a significant computational saving stemming from the dataset size reduction. As an example, using the proposed sample selection algorithm and values of l_d in the order of 1-3 metres caused reductions of more than 80% in all cases, at the price of average error increases of 8, 3 and 4 cm in the basketball field, lobby and office, respectively (20%, 3% and 4% of relative increase). However, it leaves it to the designer to decide the value of l_d for the sampling selection algorithm. On the other hand, incorporating the autocorrelation to the model eliminates the need for manually selecting any parameter, and can result in a decrease of the average localization error.

Adapting the OLS, GLS and IGLS localization methods so that the correlation is taken into account resulted in generally more biased estimates with respect to using the IID errors assumption, issue that will be investigated in later research.

Chapter 7

Joint Localization and Model Identification

Contents

While our ultimate goal is to estimate the position of the nodes of a WSN, the localization methods used so far rely on the availability of *a priori* deterministic knowledge regarding the model parameters. In turn, a proper estimation of these parameters requires training data, which has to be collected in the target scenario.

In many practical situations it is neither possible nor practical to collect such training data, and therefore the model identification prior to the production phase is not possible. This is often the case when the network has to be deployed *ad-hoc* (e.g. in emergency and rescue scenarios). Moreover, the model parameters can vary significantly depending on the HW particularities of the nodes that make the observations and their position in obstructed environments (see Chapter 3); this variability can negatively affect the performance of localization algorithms (see Chapter 5). It is then desirable to develop localization methods that do not require a training phase and that can automatically adapt to diverse environments and HW variability. In our case, this translates into a need for methods that do not rely on having pre-identified models.

This chapter is dedicated to methods that jointly estimate the position and model parameters directly based on the same vector of observations. This approach brings along two major advantages. First, it eliminates the dependency on training data. Second, because the model parameters are variables themselves, they can be automatically adjusted so that the resulting model better explains the particularities of the '*production*' data observed, rather than the training data. The models can then adapt to minor HW variabilities and local environmental inhomogeneities, resulting in better position estimates. The drawback is that, because there are more variables to be estimated, there is a higher risk of the appearance of identifiability issues, local maxima and other difficulties that will need to be addressed.

The chapter begins in Section 7.1 with a review of the existing literature on localization methods that do not rely on explicit knowledge about the model parameters. Of special importance is the one that exploits the principle of separability of least squares to concentrate the log-likelihood, to which more attention will be paid. In Section 7.2 we redefine the performance metric by which the localization algorithms will be graded when they are used with our experimental data. This redefinition is necessary due to the differences between simple localization and joint localization and model identification. Sections 7.3, 7.5 and 7.4 present the different localization algorithms proposed in the chapter. The first one is a direct global optimization of the log-likelihood that follows a convenient two-step recursion. The second and third algorithms reduce the dimensionality of the search space prior to the optimization phase by concentrating the log-likelihood. Additionally, in order to tackle the identifiability and sensitivity problems observed, we propose using MAP instead of ML estimators. Finally, Section 7.6 summarizes the main conclusions of the chapter.

7.1 Related Work

The problem of localization when the model parameters are not known has been already addressed in the extant literature, and diverse techniques for solving it have been proposed. This section presents the related work grouped in two subsections: studies aiming at a direct joint global optimization over position and model parameters and studies that eliminate the dependence of the problem on some of the model parameters.

7.1.1 Global Optimization

Both the model identification (estimation of $\boldsymbol{\theta}$) and node localization (estimation of $\vec{\boldsymbol{s}}$) are often posed as optimization problems in which the aim is to minimize a cost function. Under the log-normal model, the RSS is linear on n and z_0, and therefore these parameters can be estimated using LLS based on field measurements knowing the transmitter and receiver positions. The normality assumption of the errors under the log-normal model causes the LSE and ML criteria to coincide. Therefore, should we decide to use ML for position estimation, the same cost function is used to estimate both $\boldsymbol{\theta}$ and $\vec{\boldsymbol{s}}$: the log-likelihood. A natural approach for a joint localization and model identification is then to maximize the log-likelihood jointly over $\boldsymbol{\theta}$ and $\vec{\boldsymbol{s}}$, that is:

$$\begin{pmatrix} \hat{\vec{\boldsymbol{s}}} \\ \hat{\boldsymbol{\theta}} \end{pmatrix} = \underset{\vec{\boldsymbol{s}},\boldsymbol{\theta}}{\operatorname{argmax}} L(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}).$$

There are several problems associated with this approach. First, an iterative solver is needed to find the maximum, and therefore we need an initial solution to start the solver. This is also an issue when using ML for plain localization. However, this problem is now emphasized by the fact that we need initial estimates of both $\vec{\boldsymbol{s}}$ and $\boldsymbol{\theta}$ at the same time. Because the dimension of the input space has increased, the complexity of the cost function also increases, which now has more potential problems with local maxima. Additionally, because we have more parameters to estimate, it is more likely that the problem will be affected by identifiability issues. The proper selection of the initial guess then acquires special importance. Nevertheless, different flavours for directly optimizing the cost function have been studied.

Mailaender has conducted a simulation study on the convergence of different solvers for localization when using RSS under the log-normal model with constant autocorrelated noise [Mailaender, 2012]. He assumed that the reference power z_0 is known a priori,[1] while the path-loss exponent, noise variance and correlation

[1] Mailaender assumed that the reference power depends only on factors whose contributions to the link budget are known (antenna gains, transmitting power, etc.)

coefficient are considered nuisance parameters. Mailaender poses the localization as an optimization problem using LS and weighted least squares (WLS) criteria, resulting in non-linear problems that he solved using steepest descent and Levenverg-Marquardt methods. He first tried jointly estimating the position and all the nuisance parameters, approach that he quickly discarded due to convergence issues. He then tried to estimate the position jointly with only some of the nuisance parameters, assuming the others known *a priori*. No issues were commented regarding the sensitivity of the proposed solutions to the geometrical configuration of the problem neither to the initial position given to the solver, which was fixed.[2]

[Ferris et al., 2006] uses GPR with a squared exponential autocorrelation function to model the RSS and calculates the likelihood of the measurements as the posterior conditioned on the training data and the model hyper-parameters.[3] This likelihood is then used in a Bayesian filter together with a motion model to track a mobile device. The training data consists of RSS vectors gathered by a mobile device in known locations. The location of the fixed transmitters is, however, not known; these are treated as part of the hyper-parameters that need to be estimated from the training data[4]. All of the hyper-parameters are calculated jointly using ML and a large-scale BFGS (L-BFGS) solver. To avoid local maxima, the authors started the solver at multiple randomly selected points. No convergence issues were reported.

In [Gholami et al., 2013] the authors acknowledge the high non convexity of the log-likelihood function when the path-loss exponent and reference power are unknown, and propose some approximations that transform the original cost function into a smoother one leading to a trust region subproblem that can be solved efficiently. Using simulations they show that the resulting solution has larger RMSE than ML, but can be effectively used as an initial estimate for the ML algorithm.

Instead of a direct global optimization, Zemek *et al.* proposed using a recursion consisting of steps in [Zemek et al., 2007] and [Zemek et al., 2008]. The procedure starts with an initial guess at the model parameters, $\hat{\boldsymbol{\theta}}_0$. In the first

[2]In relation to our experience, we believe that the simulation environment used by Mailaender were very favourable for the convergence of algorithms, which is the main topic under study in the publication. First, the problem had an almost perfectly symmetrical geometry, with the beacons placed in a grid around the target node. And second, we speculate that the starting position used by the solvers was rather close to the origin given the scale of the simulated setting.

[3]The term *model hyper-parameters* is taken from [Rasmussen and Williams, 2006], where it is used to emphasize that they are parameters of a non parametric model.

[4] This is, in essence, the same problem that we are tackling in this chapter: the joint estimation of the position of our nodes and the model parameters. Ferris *et al.*, however, do not use the log-distance to model the mean of the RSS, but a simple inverse relation of the form $z = b + m\,||\vec{s} - \vec{r}||$. He then relies on the flexibility of GPR to accommodate the residuals.

step of the recursion, a position estimate $\hat{\vec{s}}_0$ is calculated taking $\hat{\boldsymbol{\theta}}_0$ as the true model parameters. In the second step $\hat{\vec{s}}_0$ is considered as the true position and used to re-estimate the model parameters. A new iteration then begins, and the recursion continues until a convergence criterion is satisfied. The authors tried different combinations of LS and ML localization and model identification methods, concluding that using ML in both gives better results. We used the same approach in [Vallet et al., 2012] applying it to the research context of this thesis. Section 7.3 gives the relevant details of this work.

7.1.2 Input Space Dimension Reduction

Given that the parameter of interest is the position, the vector $\boldsymbol{\theta}$ can be considered a nuisance parameter.[5] Several strategies have been proposed to reduce or eliminate the dependency of the cost function on nuisance parameters.

Separability of LS

One strategy for reducing the dimensionality of the search space is to exploit the principle of separability that some NLS problems exhibit [Kay, 1993, Sec. 8.9]. Under the log-normal model and the IID errors assumption, the problem of jointly estimating position and model parameters possess this property. This allows us to eliminate the dependency of the log-likelihood function on $\boldsymbol{\theta}_{\mathrm{m}} = (-10n, z_0)^t$ and σ as follows.

From (6.17) we can see that the log-normal model predicts the mean of the RSS measurements as

$$\bar{\boldsymbol{z}} = \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}}.$$

During an experiment, a vector of RSS measurements $\boldsymbol{z} = (z_1, \ldots, z_{\mathrm{K}})^t$ is collected. In order to estimate $\vec{\boldsymbol{s}}$ and $\boldsymbol{\theta}_{\mathrm{m}}$ we minimize the LS criterion

$$J(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}}) = (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}})^t \, (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}}) \tag{7.1}$$

jointly over both, $\vec{\boldsymbol{s}}$ and $\boldsymbol{\theta}_{\mathrm{m}}$. Whereas $\bar{\boldsymbol{z}}$ depends non-linearly in $\vec{\boldsymbol{s}}$ through the design matrix $\boldsymbol{A}$, it is clearly linear in $\boldsymbol{\theta}_{\mathrm{m}}$ Then, it is possible to optimize analytically $J(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}})$ over $\boldsymbol{\theta}_{\mathrm{m}}$ by finding its value such that the gradient becomes zero. This leads to the classical OLS estimator

$$\hat{\boldsymbol{\theta}}_{\mathrm{m}}(\vec{\boldsymbol{s}}) = (\boldsymbol{A}^t\boldsymbol{A})^{-1}\boldsymbol{A}^t \, \boldsymbol{z},$$

which is an expression that depends only on $\vec{\boldsymbol{s}}$. Substituting $\hat{\boldsymbol{\theta}}_{\mathrm{m}}(\vec{\boldsymbol{s}})$ for $\boldsymbol{\theta}_{\mathrm{m}}$ in (7.1) yields

$$J(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}) = (\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}})^t(\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}}), \tag{7.2}$$

[5] Parameters that are necessary to know, but that are not of primary importance.

where the dependency on $\boldsymbol{\theta}_{\mathrm{m}}$ has been eliminated. Minimizing (7.2) with respect to $\vec{\boldsymbol{s}}$ yields the NLS estimator of the node's position

$$\hat{\vec{\boldsymbol{s}}} = \underset{\vec{\boldsymbol{s}}}{\operatorname{argmax}}\, J\left(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}(\vec{\boldsymbol{s}})\right). \tag{7.3}$$

Note that the criterion (7.2) depends only on $\vec{\boldsymbol{s}}$, and cannot be used to estimate the parameters of the covariance model $\boldsymbol{\theta}_{\mathrm{c}}$. This comes at no surprise, as the LS criterion does not use any information about the PDF of the data. In order to estimate $\boldsymbol{\theta}_{\mathrm{c}}$ we can use the ML criterion.

Under the IID normally distributed errors assumption, (7.3) is also the ML estimator. The vector of parameters $\boldsymbol{\theta}_{\mathrm{c}}$ is reduced to $\boldsymbol{\theta}_{\mathrm{c}} = \sigma = \sigma_c$, and from (5.1) and using (7.2), the log-likelihood can be expressed as:

$$L(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}, \sigma^2) = -\frac{K}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} J(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}), \tag{7.4}$$

where the dependency on $\boldsymbol{\theta}_{\mathrm{m}}$ has already been eliminated through the use of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ instead. The dependency of (7.4) on σ can be further eliminated by optimizing over σ^2, which yields the following ML estimator:

$$\widehat{\sigma^2}(\vec{\boldsymbol{s}}) = \frac{1}{K} J(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}). \tag{7.5}$$

Substituting $\widehat{\sigma^2}$ for σ^2 in (7.4) yields

$$\begin{aligned} L(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}(\vec{\boldsymbol{s}}), \widehat{\sigma^2}(\vec{\boldsymbol{s}})) &= -\frac{K}{2}\ln(2\pi) - \frac{K}{2}\ln\left(\frac{1}{K} J(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}})\right) - \frac{K}{2} \\ &= -\frac{K}{2}\left[1 + \ln(2\pi) - \ln K\right] - \frac{K}{2}\ln J(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}), \end{aligned} \tag{7.6}$$

which is a function that depends only on $\vec{\boldsymbol{s}}$. Note that (7.6) depends on $J(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}(\vec{\boldsymbol{s}}))$ via the logarithm. Therefore, maximizing one yields the same solution as maximizing the other. This makes clear once more the equivalence of the LS and ML criteria. Once the estimate $\hat{\vec{\boldsymbol{s}}}$ has been calculated, we can calculate $\widehat{\sigma^2}$ by substituting $\hat{\vec{\boldsymbol{s}}}$ for $\vec{\boldsymbol{s}}$ in (7.5).

This technique of reducing the number of parameters of the log-likelihood function is called *profiling* by statisticians and *concentration of the log-likelihood* by econometricians. The likelihood is then said to be concentrated with respect to the parameters whose dependence has been eliminated [Davidson and MacKinnon, 2004, ch. 10]. In the following we will borrow this terminology. Thus, (7.6) represents the log-likelihood concentrated with respect to $\boldsymbol{\theta}_{\mathrm{m}}$ and σ.

A similar technique for reducing the dimensionality of the search space has been used for example in [Gustafsson and Gunnarsson, 2007], and later in [Gustafsson et al., 2012]. In their initial work Gustafsson *et al.* present a framework

for localization and tracking in WSNs using information from different groups of sensors that measure energy-related signals for which the log-distance model is adequate, such as RSS, seismic, acoustic and so forth. Each group has its own model parameters that optimally describe the particular type of observations. The measurements are all assumed to be IID, and the principle of separability is used to eliminate the nuisance parameters similarly as we have described previously. The main difference is that, in order to fuse the information from sensors of different nature, the authors use the WLS instead of the LS criterion to weight the measurements coming from the different type of sensors. In [Gustafsson et al., 2012] the authors present more detailed results from real experiments.

[Salman et al., 2012] assume that the reference power is known. They study and compare using simulations different localization algorithms: one that maximizes the log-likelihood, another maximizing the concentrated log-likelihood and a third one following a MAP approach, all of them considering the path-loss exponent as a nuisance parameter. The proposed MAP estimator is of the form

$$\hat{\vec{\boldsymbol{s}}}_{\text{MAP}} = \underset{\vec{\boldsymbol{s}}}{\operatorname{argmax}}\, p(\vec{\boldsymbol{s}}, n|\boldsymbol{z}) = \underset{\vec{\boldsymbol{s}}}{\operatorname{argmax}}\, p(\boldsymbol{z}|\vec{\boldsymbol{s}}, n)p(n),$$

where the prior for the path-loss exponent, $p(n)$, is a Gaussian centred at the true value of n. In their simulations, optimizing the log-likelihood and the concentrated log-likelihood yielded the same results. No convergence issues were reported.[6] The authors also show that the MAP estimator has larger RMSE than ML.

Marginalization

Another well-known technique for eliminating the dependency on nuisance parameters is marginalization. Should a prior PDF of the model parameters, $p(\boldsymbol{\theta})$, be available, we can calculate the conditional likelihood of $\boldsymbol{z}$ given $\vec{\boldsymbol{s}}$ as follows:

$$p(\boldsymbol{z}|\vec{\boldsymbol{s}}) = \int_{S_{\boldsymbol{\theta}}} p(\boldsymbol{z}|\vec{\boldsymbol{s}}, \boldsymbol{\theta})p(\boldsymbol{\theta}|\vec{\boldsymbol{s}})d\boldsymbol{\theta} = \int_{S_{\boldsymbol{\theta}}} p(\boldsymbol{z}|\vec{\boldsymbol{s}}, \boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{7.7}$$

where $S_{\boldsymbol{\theta}}$ is the support of $\boldsymbol{\theta}$, and $p(\boldsymbol{\theta}|\vec{\boldsymbol{s}}) = p(\boldsymbol{\theta})$ is implicitly assumed. The conditional likelihood (7.7) does not depend any longer on $\boldsymbol{\theta}$, and an estimate of $\vec{\boldsymbol{s}}$ can be found by maximizing the conditional log-likelihood as usual. In practice, typically it is not possible to do the integration analytically. In that case one can rely on Monte Carlo integration methods, in which the integral is substituted by

[6] Based on our experience, we believe that the simulation environment is very advantageous for the convergence of the algorithm, with 4, 5 and 6 reference nodes placed in a circle of 50 m radius, the target node in the centre of the circle and the initial position at (25,25).

a summation and the prior $p(\boldsymbol{\theta})$ is represented by a set of samples. In that case (7.7) becomes

$$p(\boldsymbol{z}|\vec{\boldsymbol{s}}) = \sum_{S_{\boldsymbol{\theta}}} p(\boldsymbol{z}|\vec{\boldsymbol{s}}, \boldsymbol{\theta}) p(\boldsymbol{\theta}),$$

where $S_{\boldsymbol{\theta}}$ is now the set of samples that represent the prior $p(\boldsymbol{\theta})$. This technique has been used, for example, in [Letchner et al., 2005], in which the authors propose a hierarchical Bayesian model for WLAN localization.

7.2 Performance Metric

In the remainder of this chapter we will present the results of applying different localization algorithms to the datasets of our experiments. Therefore it is necessary to define a measure to grade their performance when using our experimental data. We already presented such a measure in Chapter 5 (see (5.5)). However, this measure was developed specifically to compare common and individual models based on deterministic knowledge of the model parameters, and was specifically designed to avoid using the same observations in the localization and model identification phases. In this chapter, both position and model parameters are estimated using the same vector of measurements. We then need to define a new performance metric with which we can grade different localization algorithms.

Recall that for each environment we have three datasets per node (one per robot trajectory) for a total of M nodes available. Therefore, we have in total $3 \times M$ datasets that we can use to jointly estimate their respective $\vec{\boldsymbol{s}}$ and $\boldsymbol{\theta}$. Our measure of performance is simply the mean error of all possible estimation outcomes:

$$e = \frac{1}{3 \cdot M} \sum_{j=1}^{3} \left(\sum_{i=1}^{M} \left\| \vec{\boldsymbol{s}}_{\mathrm{i}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{ij}} \right\| \right), \tag{7.8}$$

where $\vec{\boldsymbol{s}}_{\mathrm{i}}$ is the true position of the i-th node and $\hat{\vec{\boldsymbol{s}}}_{\mathrm{ij}}$ its position estimate when using the data collected during the j-th robot trajectory.

7.3 Localization-Model Identification Recursion

The work presented in this section is our first practical approach to the localization problem without having knowledge of the model parameters. It aims at a direct global optimization of the log-likelihood function, and thus it requires initial estimates of both, the model parameters and the target node's position. Finding these initial estimates is a problem in itself.

Table 7.1: Localization-model identification recursion algorithm (LMIR).

```
1 lmir(z, R, θ̂_0)
2   θ̂ = θ̂_0
3   repeat
4     ŝ⃗=locate_node(z, R, θ̂)
5     d̂=calculate_robot_node_distances(ŝ⃗, R)
6     θ̂=extract_rss_model(z, d̂)
7   until convergence
8   return ŝ⃗, θ̂
```

Should we have available an initial value of either $\hat{\vec{\boldsymbol{s}}}_0$ or $\hat{\boldsymbol{\theta}}_0$, the other one can be estimated taking the first one as the true value. For example, suppose that we have available $\hat{\boldsymbol{\theta}}_0$: we can then find $\hat{\vec{\boldsymbol{s}}}_0$ using ML localization taking $\hat{\vec{\boldsymbol{s}}}_0$ as the true model parameters. On the other hand, should we have $\hat{\vec{\boldsymbol{s}}}_0$, we can estimate $\hat{\boldsymbol{\theta}}_0$ using OLS taking $\hat{\vec{\boldsymbol{s}}}_0$ as the true node position.

Inspired by this idea, we proposed in [Vallet et al., 2012][7] the localization method illustrated in Table 7.1. The algorithm accepts as inputs the set of observations, $\boldsymbol{z}$, the trajectory of the robot, $\boldsymbol{R}$, and the initial guess at the model parameters, $\hat{\boldsymbol{\theta}}_0$. In line 4, the position of the node is estimated taking $\hat{\boldsymbol{\theta}}$ as the true value of the model parameters. This can be done by e.g. using ML estimation. In the first iteration $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}_0$, and therefore the result is an initial position estimate. In line 5, the node-robot distance estimates, $\hat{\boldsymbol{d}}$, are calculated using $\hat{\vec{\boldsymbol{s}}}$ as the true node position. Finally, line 6 re-estimates $\hat{\boldsymbol{\theta}}$ using $\hat{\boldsymbol{d}}$ as the true node-robot distances. This procedure goes back to line 4, in which the new value of $\hat{\boldsymbol{\theta}}$ is used as the true model parameters. The iteration then continues until a suitable convergence criterion is satisfied (e.g. the distance between $\hat{\vec{\boldsymbol{s}}}$ in consecutive iterations is smaller than a certain δ). The returned values are the final values of $\hat{\boldsymbol{\theta}}$ and $\hat{\vec{\boldsymbol{s}}}$. In the following we will refer to this algorithm as LMIR.

Note that we could have formulated a reciprocal algorithm starting from an initial position estimate $\hat{\vec{\boldsymbol{s}}}_0$ instead of $\hat{\boldsymbol{\theta}}_0$. However, it can be argued that finding initial estimates of $\boldsymbol{\theta}$ is easier than for $\vec{\boldsymbol{s}}$ in practical *ad-hoc* deployments in which prior knowledge of the environment is very limited or non-existent. In principle, the range of possible initial values for $\vec{\boldsymbol{s}}$ is unlimited. One can of course exploit prior knowledge to find a region of space in which the node can lay, for example by somehow defining a region in which the node-robot communication is physically possible. In any case, once the feasibility region has been defined, the problem

[7] After our publication we found out that the same approach had been already proposed earlier by Zemek *et al.* in [Zemek et al., 2007] and [Zemek et al., 2008].

of finding a good initial position estimate within that area still remains.

Regarding the initial estimate of the model parameters, the situation is slightly different. In this case, the range of possible values is naturally more limited. For example, we know that in order for the log-normal model to be plausible the path-loss exponent should be positive and the reference power negative. Furthermore, we know typical values for different types of environments from studies in the literature. A key issue here is that these particular parameter values characterize types of environments, that is: the same guess at the model parameters can be used in many particular scenarios as long as the scenarios are similar regarding their complexity. Because of these considerations, we chose an implementation of the LMIR algorithm that starts with an initial guess at the model parameters.

One of the advantages of the LMIR algorithm is that its implementation is trivial once we have available any localization and model identification algorithms. We already have these algorithms for the two families of generic polynomial RSS models described in Section 5.6. In this section we will use the LMIR algorithm with all these models.[8] The underlying idea is that these generic models are more flexible than the traditional log-normal model, and can accommodate better the typical non-linearities and deviations of the residuals with respect to the normality assumption that we observed and commented on Section 5.6. Because of this, one naturally expects that using such models will yield better results in the context of joint localization and model identification.

The generic models presented in Section 5.6 can be divided based on whether their errors follow either a Gaussian or an MoG distributions. In either case, we estimate $\vec{\boldsymbol{s}}$ using ML in the LMIR algorithm. Regarding the model identification, in the case of Gaussian errors we estimate $\boldsymbol{\theta}$ using OLS, and in the case of MoG we use expectation maximization (EM) (see section 5.6 for details). Note that both steps, localization and model identification, maximize the log-likelihood in their own respective subspace. Therefore, the procedure shown in Table 7.1 is itself a maximization of the log-likelihood jointly over $\vec{\boldsymbol{s}}$ and $\boldsymbol{\theta}$. The difference with respect to using a traditional gradient-based solver is that the optimization is done in consecutive steps in two orthogonal sub-spaces: the sub-spaces of positions and model parameters. Anyway, because the same function is being optimized in both steps, in our case the algorithm is guaranteed to increase the log-likelihood in every new iteration.

We will now compare the localization performance of LMIR using our generic polynomial RSS-distance models. In order to assess the best possible accuracy of the position estimates that we can achieve in practice when jointly estimating $\vec{\boldsymbol{s}}$ and $\boldsymbol{\theta}$ using our datasets, we conduct a preliminary analysis in which we estimate the position using ML given the value of the model parameters estimated assuming perfect knowledge of $\vec{\boldsymbol{s}}$. That is: for each dataset we first fit the RSS model

[8] Note that they all assume IID errors

Table 7.2: Experimental lower bounds of the position error (distance in metres) for ML-based joint localization and model identification methods

	P1LM1N	**P1LM2N**
Bfield	0.37	0.34
Lobby	0.64	0.54
Office	0.78	0.69
	P3LM1N	**P3LM2N**
Bfield	0.36	0.33
Lobby	0.68	0.53
Office	0.59	0.58
	P3M1N	**P3M2N**
Bfield	0.41	0.40
Lobby	0.70	0.57
Office	0.59	0.58

Table 7.3: Average errors of the position estimates calculated with the algorithm from Table 7.1 in different environments using log-normal and P3M2N models.

	P1LM1N				**P3M2N**			
	min	mean	median	max	min	mean	median	max
Bfield	0.05	0.64	0.30	4.84	0.09	0.47	0.32	2.44
Lobby	0.11	1.27	0.79	6.17	0.12	1.06	0.59	5.04
Office	0.16	1.18	0.93	2.72	0.16	0.97	0.76	2.32

using the true $\vec{s}$. Then, using the same dataset, we estimate the position using the previous model. Table 7.2 presents the resulting average errors calculated as in (7.8) when using the different polynomial models being considered (see Section 5.6 for the explanation of the nomenclature). In all cases we used GS to find the global maxima with a 0.1 m grid resolution.

As we can see, the largest significant change in performance occurs in the office when increasing the polynomial degree from 1 to 3 (P1LM1N to P3LM1N). This is our most complex and inhomogeneous space, where significant changes in the environment can occur in short relative distances. It is in these types of environments where using more complex models can be more advantageous, as

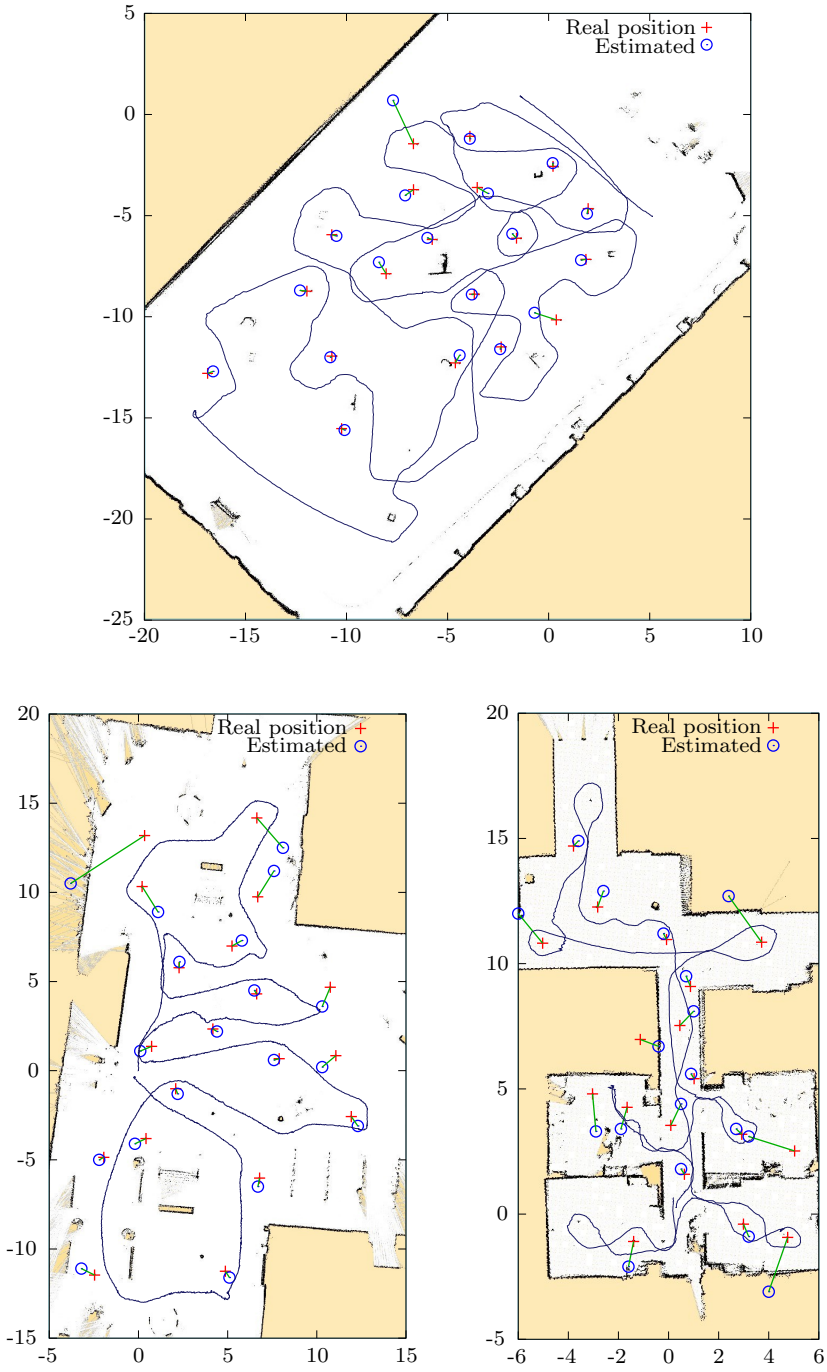


Figure 7.1: Maps generated by the robot, real position of the nodes and position estimates using the algorithm from Table 7.1 with P3M2N models using data from one robot trajectory in three different environments. The trajectories of the robot are represented as dark blue continuous lines. Coordinates are in metres.

it is also where one expects stronger deviations with respect to the log-normal model. This local specialization results in better position estimates. Using MoGs contribute to a more specialization of the model, resulting in a further general decrease in the mean error.

Table 7.3 presents a comparison of the results when using the LMIR algorithm for localization in our experimental setup. Results are presented for the log-normal and the P3M2N models. The results for the P3LM2N model are similar to those for the P3M2N model, and are omitted for brevity. In all cases we use GS to maximize the log-likelihood (line 4 in Table 7.1) with a 0.1 m grid resolution and either OLS or EM for the model identification as explained previously (line 6 in Table 7.1). The initial value of the model parameters, $\boldsymbol{\theta}_0$, is the same for all the nodes within the same environment. These are chosen as the median of the parameters from the individual models identified in Section 3.2. As we can see, the mean errors are larger than the lower limits presented in Table 7.2, but within the same range as those corresponding to using IMs in a simple localization scenario with $\boldsymbol{\theta}_{\mathrm{m}}$ estimated a priori (see results for IM in Table 5.6). Finally, Figure 7.1 presents sample localization results when using P3M2N models and data collected from a single robot trajectory in each environment.

One important issue that is worthy of further attention is that we noticed a high sensitivity of the results to variations in the initial values $\boldsymbol{\theta}_0$. In some cases the localization errors were in the order of tens of metres and even more, and this occurred as soon as $\boldsymbol{\theta}_0$ was different than the median. This high sensitivity is not specifically associated with the LMIR algorithm, but rather with the general problem of jointly maximizing the log-likelihood over the position and model parameters. To our surprise, this issue has not received significant attention in prior studies. We address this problem in the following sections.

7.4 Concentrated Log-likelihood With Respect to the Model Parameters

In Subsection 7.1.2 we reviewed the technique of concentrating of the log-likelihood with respect to the model parameters under the log-normal model with IID measurements. In this section we first extend the results to the case when using the model 6.17, which considers the spatial correlation among samples. This new concentrated log-likelihood will be used to locate the nodes in our experimental environments. Most of the results presented in this section have been published in [Vallet García, 2015].

The log-likelihood of a set of measurements under the model 6.17 is

$$L(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}}, \boldsymbol{\theta}_{\mathrm{c}}) = -\frac{1}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} J_{\Sigma}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}}, \boldsymbol{\theta}_{\mathrm{c}}) - \frac{K}{2} \ln (2\pi), \tag{7.9}$$

where

$$J_{\Sigma}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}}, \boldsymbol{\theta}_{\mathrm{c}}) \equiv (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}})^{t}\, \boldsymbol{\Sigma}^{-1}\, (\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}}) \tag{7.10}$$

is the GLS criterion. Equally as in the case when assuming IID measurements, the vector $\bar{\boldsymbol{z}}$ depends linearly on the model parameters $\boldsymbol{\theta}_{\mathrm{m}}$, whereas it does not in $\boldsymbol{\theta}_{\mathrm{c}}$ neither in $\vec{\boldsymbol{s}}$. The problem then exhibits the property of separability of least squares, and we can find a linear estimator of $\boldsymbol{\theta}_{\mathrm{m}}$ that depends on the rest of the parameters. Maximizing (7.10) over $\boldsymbol{\theta}_{\mathrm{m}}$ leads to the GLS estimator

$$\hat{\boldsymbol{\theta}}_{\mathrm{m}}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{c}}) = (\boldsymbol{A}^{t}\boldsymbol{\Sigma}^{-1}\boldsymbol{A}^{t})^{-1}\boldsymbol{A}^{t}\boldsymbol{\Sigma}^{-1}\boldsymbol{z}. \tag{7.11}$$

Substituting $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ for $\boldsymbol{\theta}_{\mathrm{m}}$ in (7.10) we obtain

$$J_{m}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{c}}) \equiv J_{\Sigma}(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}, \boldsymbol{\theta}_{\mathrm{c}}) = (\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}})^{t}\boldsymbol{\Sigma}^{-1}(\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}}), \tag{7.12}$$

which leads to the following concentrated log-likelihood:

$$L_{m}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{c}}) \equiv L(\boldsymbol{z}; \vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}, \boldsymbol{\theta}_{\mathrm{c}}) = -\frac{1}{2}\left[\ln|\boldsymbol{\Sigma}| + J_{\mathrm{m}}(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{c}}) + K\ln(2\pi)\right], \tag{7.13}$$

where the dependency on $\boldsymbol{\theta}_{\mathrm{m}}$ has been eliminated. The new concentrated log-likelihood (7.13) now depends only on $\vec{\boldsymbol{s}} = (x, y)^{t}$ and $\boldsymbol{\theta}_{\mathrm{c}} = (\sigma_n, \sigma_c, l_0)^{t}$.

In the case of having IID measurements, the dependence on σ could still be eliminated using its ML estimator. To find it we simply had to calculate the derivate of the log-likelihood (7.4) with respect to σ^2 and find the value that makes it zero. Calculating this derivative was straightforward because σ^2 appeared in the log-likelihood factored out of the term J_{Σ}. However, when considering autocorrelated observations, the log-likelihood depends on $\boldsymbol{\theta}_{\mathrm{c}}$ via the covariance $\boldsymbol{\Sigma} = \sigma_n^2\mathbf{I} + \sigma_c^2\boldsymbol{W}$, which affects the log-likelihood through the logarithm of its determinant and the quadratic form J_{Σ}, in which it is inverted. Henceforth, finding analytical expressions for σ_n, σ_c or l_0 that maximize the log-likelihood is not straightforward.

It is still possible to reduce further the dimensionality of the input space using a simple re-parameterization of the model. Let us call

$$\begin{aligned} \sigma^2 &= \sigma_n^2 + \sigma_c^2 \\ \alpha &= \sigma_n^2/\sigma^2. \end{aligned}$$

Then we have that

$$\begin{aligned} \sigma_n^2 &= \alpha\sigma^2 \\ \sigma_c^2 &= (1-\alpha)\sigma^2, \end{aligned} \tag{7.14}$$

where $0 \leq \alpha \leq 1$. This is in fact a parameterization that we have already used in the previous chapter. Using the new parameters, the covariance matrix can be expressed as

$$\boldsymbol{\Sigma} = \sigma_n^2 \mathbf{I} + \sigma_c^2 \boldsymbol{W} = \alpha\sigma^2 \mathbf{I} + (1-\alpha)\sigma^2 \boldsymbol{W} = \sigma^2 \left[\alpha \mathbf{I} + (1-\alpha)\boldsymbol{W}\right] = \sigma^2 \boldsymbol{\Phi},$$

where

$$\boldsymbol{\Phi}(\alpha, l_0) \equiv \alpha \mathbf{I} + (1-\alpha)\boldsymbol{W}. \tag{7.15}$$

Then, we find that

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma^2}\boldsymbol{\Phi}^{-1}, \;\; |\boldsymbol{\Sigma}| = \sigma^{2K}|\boldsymbol{\Phi}|,$$

with which we can rewrite (7.12) as

$$J_m(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{c}}) = \frac{1}{\sigma^2}(\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}})^t \boldsymbol{\Phi}^{-1}(\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}}) = \frac{1}{\sigma^2} J_\Phi, \tag{7.16}$$

where

$$J_\Phi(\vec{\boldsymbol{s}}, \alpha, l_0) \equiv (\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}})^t \boldsymbol{\Phi}^{-1}(\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_{\mathrm{m}}). \tag{7.17}$$

Using this, the concentrated log-likelihood (7.13) becomes

$$\begin{aligned} L_m(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{c}}) &= -\frac{1}{2}\ln\left(\sigma^{2K}|\boldsymbol{\Phi}|\right) - \frac{1}{2\sigma^2} J_\Phi - \frac{K}{2}\ln(2\pi) \\ &= -\frac{K}{2}\ln\sigma^2 - \frac{1}{2}\ln|\boldsymbol{\Phi}| - \frac{1}{2\sigma^2} J_\Phi - \frac{K}{2}\ln(2\pi). \end{aligned} \tag{7.18}$$

The GLS estimator $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ from (7.11) also becomes

$$\hat{\boldsymbol{\theta}}_{\mathrm{m}} = (\boldsymbol{A}^t \frac{1}{\sigma^2}\boldsymbol{\Phi}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^t \frac{1}{\sigma^2}\boldsymbol{\Phi}^{-1}\boldsymbol{z} = (\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\boldsymbol{z}, \tag{7.19}$$

and therefore it does not depend on σ^2.

Finding the value of σ^2 that minimizes (7.18) is straightforward, leading to

$$\widehat{\sigma^2} = \frac{1}{K} J_\Phi. \tag{7.20}$$

Finally, substituting $\widehat{\sigma^2}$ for σ^2 in (7.18) yields

$$\begin{aligned} L_{mc}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \alpha, l_0) &= L_m(\boldsymbol{z}; \vec{\boldsymbol{s}}, \hat{\sigma}, \alpha, l_0) \\ &= -\frac{K}{2}(1 + \ln 2\pi) - \frac{K}{2}\ln\left(\frac{1}{K} J_\Phi\right) - \frac{1}{2}\ln|\Phi| \\ &= -\frac{K}{2}\left[1 + \ln 2\pi + \ln\left(\frac{J_\Phi}{K}\right)\right] - \frac{1}{2}\ln|\Phi|, \end{aligned} \tag{7.21}$$

which does not depend on σ. Optimizing over the remaining parameters, $\vec{\boldsymbol{s}}$, α and l_0, can be done as usual with a gradient-based solver, for which it is convenient to calculate the derivatives analytically. Appendix C.1 presents the details of the tedious, but convenient, derivation of the analytical formulas for gradient.

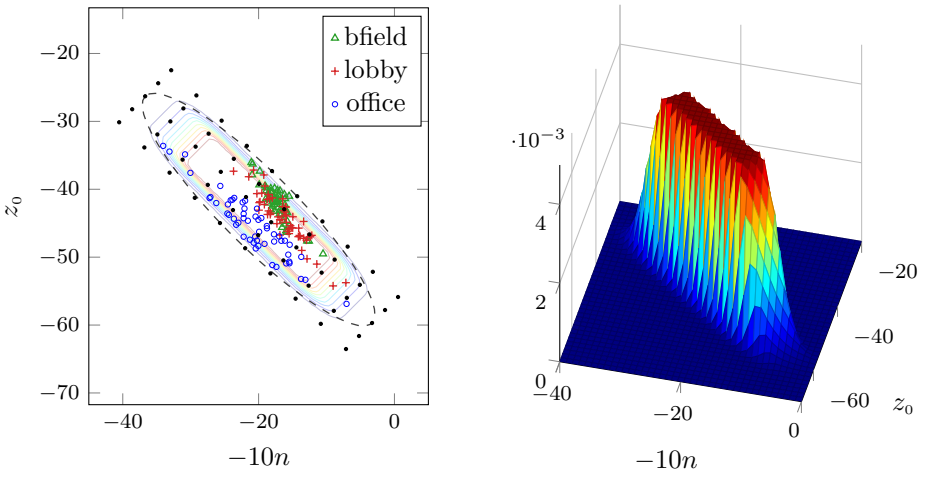
7.4.1 Model Identification

Estimating position is typically done once that the model parameters are known. These are in turn estimated *a priori* from training data knowing the T-R distances corresponding to each RSS measurement (i.e. knowing $\vec{\boldsymbol{s}}$). Once estimated, the same vector of model parameters is used to locate many nodes many times in the given environment, each time using a different set of observations. Therefore, the data used for position estimation is different than that used for model identification, with the latter being always the same.

The problem of localization is different with respect to jointly estimating the position and model parameters in that both have to be inferred using the same dataset. For each realization of the observations $\boldsymbol{z}$ we will obtain a different position and model parameters estimates. Therefore, the model parameters are not fixed deterministic values anymore, but rather random variables conditioned on the experimental setup (HW used and type of environment). The goal of this subsection is to uncover this variability, which will for example be useful to assign prior distributions to the different model parameters. In particular, we are interested in the values of $\boldsymbol{\theta}_{\mathrm{m}}$, as these are the parameters that mostly affect the localization error. We will present sample values obtained by identifying IMs with all the data from our experimental setup (see Section 1.4). In this subsection, we will then assume that $\vec{\boldsymbol{s}}$ is known.

Under the typical assumption of IID errors we have that $\boldsymbol{\Sigma} = \sigma^2\mathbf{I}$. Then (7.10) becomes the LS criterion, and $\boldsymbol{\theta}_{\mathrm{m}}$ can be estimated using LLS. In the presence of correlated noise, as is our case, the OLS estimator is inefficient and GLS should be used instead [Amemiya, 1985, sec 6.1]. However, the GLS estimator of $\boldsymbol{\theta}_{\mathrm{m}}$ (which minimizes (7.10) and therefore maximizes (7.9)) requires knowing $\boldsymbol{\Sigma}$, for which $\boldsymbol{\theta}_{\mathrm{c}}$ is needed (see (7.11)). Thus, we need to jointly estimate $\boldsymbol{\theta}_{\mathrm{m}}$ and $\boldsymbol{\theta}_{\mathrm{c}}$.

Should we decide to use ML and the log-likelihood (7.9) for that matter, we would have to optimize over the five parameters within $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{c}}$. Instead, using the concentrated log-likelihood of (7.21) is especially convenient, as it requires optimizing only over α and l_0. Once calculated the estimates $\hat{\alpha}$ and $\hat{l}_0$, we can then calculate $\hat{\boldsymbol{\Phi}}$, with which we can then calculate $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ using (7.19). Finally, σ^2 can be estimated using (7.20) after calculating J_Φ from (7.17). In fact, $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ and J_Φ are quantities necessary for calculating the concentrated log-likelihood, and can be retrieved from a careful implementation without needing to re-calculate them.



(a) Estimates, priors and grid for $\boldsymbol{\theta}_{\mathrm{m}}$.

(b) 3D plot of the smooth-box prior.

Figure 7.2: Values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ calculated from our experimental data, selected priors and grid for $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ (black dots).

Table 7.4: Statistics of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$.

	mean	Cov	ρ
Basketball field	$\begin{pmatrix} -17.26 \\ -42.00 \end{pmatrix}$	$\begin{pmatrix} 3.35 & -4.08 \\ -4.08 & 6.64 \end{pmatrix}$	-0.86
Lobby	$\begin{pmatrix} -16.27 \\ -44.31 \end{pmatrix}$	$\begin{pmatrix} 9.43 & -9.92 \\ -9.92 & 12.63 \end{pmatrix}$	-0.91
Office	$\begin{pmatrix} -20.97 \\ -45.36 \end{pmatrix}$	$\begin{pmatrix} 25.26 & -21.80 \\ -21.80 & 21.70 \end{pmatrix}$	-0.93
Priors	$\begin{pmatrix} -20.00 \\ -43.00 \end{pmatrix}$	$\begin{pmatrix} 49.00 & -44.10 \\ -44.10 & 49.00 \end{pmatrix}$	-0.90

Figure 7.2 shows the values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ estimated using all the available datasets from our experiments as explained previously, together with the priors and a grid of values (black dots) that will be used later in Subsection 7.4.2 and Section 7.4, respectively. The solver used is the L-BFGS-B constrained by $0 < \alpha < 1$ and $l_0 > 0$, and using $\alpha = 0.5$ and $l_0 = 1$ as initial values in all cases. Table 7.4 presents the mean and covariance of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ calculated for each environment, together with the correlation coefficient $\rho = \mathrm{Corr}\,[-10n, z_0]$. Note that these parameter estimates are different with respect to those calculated in Chapter 3, in which OLS was used to identify the individual models.

7.4.2 MAP Localization

Position estimation can be done by maximizing the concentrated log-likelihood (7.21) jointly over $\vec{\boldsymbol{s}}$, α and l_0. However, we have ascertained experimentally that the result can be very sensitive to the initial position given to the solver and the geometry of the problem. Often the localization errors obtained are quite large and come with associated infeasible estimates $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ containing negative values of $\hat{n}$ and positive ones of $\hat{z}_0$. This happens particularly when the node to be located is outside the polygonal area enclosed by the beacons or close to its perimeter. Because the concentrated log-likelihood does not depend on $\boldsymbol{\theta}_{\mathrm{m}}$ explicitly, a direct bounded optimization limiting its values is not possible. To tackle this problem we propose using MAP adding priors on $\boldsymbol{\theta}_{\mathrm{m}}$, but in an indirect way. Instead of maximizing directly the concentrated log-likelihood (7.21), we estimate $\vec{\boldsymbol{s}}$ (jointly with α and l_0) as

$$\begin{pmatrix} \hat{\vec{\boldsymbol{s}}} & \hat{\alpha} & \hat{l}_0 \end{pmatrix}^t = \underset{\vec{\boldsymbol{s}},\alpha,l_0}{\operatorname{argmax}} \left[L_{mc}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \alpha, l_0) + L_p(\hat{\boldsymbol{\theta}}_{\mathrm{m}}) \right], \tag{7.22}$$

where $L_p(\hat{\boldsymbol{\theta}}_{\mathrm{m}})$ is the log-prior of $\boldsymbol{\theta}_{\mathrm{m}}$ evaluated at the estimate $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ as in (7.19). Therefore, the resulting function to be maximized still does not depend on $\boldsymbol{\theta}_{\mathrm{m}}$.

The log-prior L_p acts as a regularizer that penalizes positions to which correspond infeasible values of $\boldsymbol{\theta}_{\mathrm{m}}$ as estimated by $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$. Panel 7.2a shows two selected priors for which we will present results. The first one is a multivariate Gaussian, whose 95% confidence ellipse is shown with a black dashed line. The contour solid lines correspond to the second prior, which is a especially designed function that acts as a smooth box in which the parameters are to be confined. Its central area has zero slope, and therefore parameter values laying in that area will not be affected by the prior. Panel 7.2b shows a three-dimensional plot of the prior. In the following we will refer to this function as *smooth-box* prior. The values for the mean and covariance are the same for both priors (see Table 7.4), and are selected so that they englobe the parameter values observed from all of our experiments.

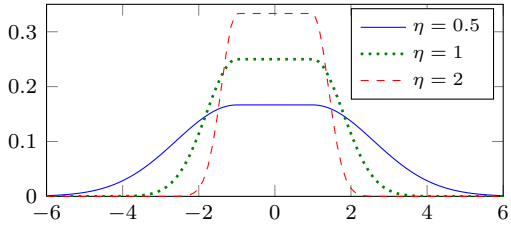


Figure 7.3: Univariate smooth box prior.

The PDF of the smooth-box prior is built from the following univariate distribution found in [Rasmussen and Nickisch, 2010]:

$$f(t;a,b,\eta) = \begin{cases} \frac{1}{\kappa}\,\exp\left\{-\frac{(t-a)^2}{2\sigma_{\mathrm{p}}^2}\right\}, & t \le a \\ \frac{1}{\kappa}, & a < t < b \\ \frac{1}{\kappa}\,\exp\left\{-\frac{(t-b)^2}{2\sigma_{\mathrm{p}}^2}\right\}, & t \ge b \end{cases} \tag{7.23}$$

where $\sigma_{\mathrm{p}} = w/(\eta\sqrt{2\pi})$, $w = |b-a|$ and $\kappa = w\,(1+\eta^{-1})$. In essence, $f(t)$ is a Gaussian cut into two parts with a uniform distribution inserted in the middle. The constant κ is a normalization factor that ensures that the function is a valid PDF. The width of the PDF is controlled with the parameters a and b, and the decay rate with η. Larger values of η make the distribution more squared, and smaller ones smoother. Figure 7.3 shows examples with $a = -1$, $b = 1$ and different values of η. Calling $q = \frac{w}{2}$, the covariance of $f(t)$ can be expressed as:

$$\mathrm{Cov}\left[f(t)\right] = \frac{\eta}{1+\eta}\left[\frac{q^2}{3} + \sigma_{\mathrm{p}}\sqrt{\frac{\pi}{2}}\left(q + \frac{\sigma_{\mathrm{p}}^2}{q}\right) + 2\sigma_{\mathrm{p}}^2\right]. \tag{7.24}$$

The multivariate smooth-box prior of Figure 7.2 (see contour lines in panels 7.2a and 7.2b) is created by multiplying two independent univariate distributions $f(-10n)f(z_0)$ with $\eta = 2$ and applying an affine transformation so that the resulting PDF has the mean and covariance of the priors specified in Table 7.4.

The effect of adding the smooth-box prior can be seen in Figure 7.4. Panel 7.4a shows the surface of the log-likelihood of the observations collected in one of our experiments, in which the model parameters are all estimated a priori. The true position of the node is marked with a plus sign, which is close to the global maximum. Panel 7.4b shows the surface of the concentrated log-likelihood for the same data. As we can see, the fact that $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ is variable creates more uncertainty. This can be seen in the figure as the appearance of a new area with large log-likelihood, together with rather flat areas around the maxima. The area with

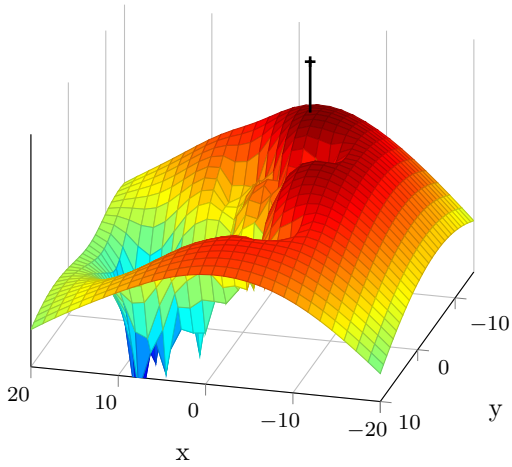
Table 7.5: Statistics of localization errors [m]

Prior	Environment	min	mean	median	max
No priors	**Basketball field**	0.03	0.61	0.29	7.12
	Lobby	0.02	1.67	0.48	18.40
	Office	0.07	1.64	0.95	8.60
Gaussian	**Basketball field**	0.03	0.53	0.29	5.14
	Lobby	0.02	0.90	0.44	7.10
	Office	0.02	1.01	0.86	3.07
Smooth box	**Basketball field**	0.02	0.53	0.29	4.66
	Lobby	0.02	0.99	0.48	6.69
	Office	0.07	0.86	0.68	2.16
Smooth box (IID)	**Basketball field**	0.03	0.55	0.32	5.49
	Lobby	0.06	1.23	0.51	7.11
	Office	0.11	1.31	1.40	2.83

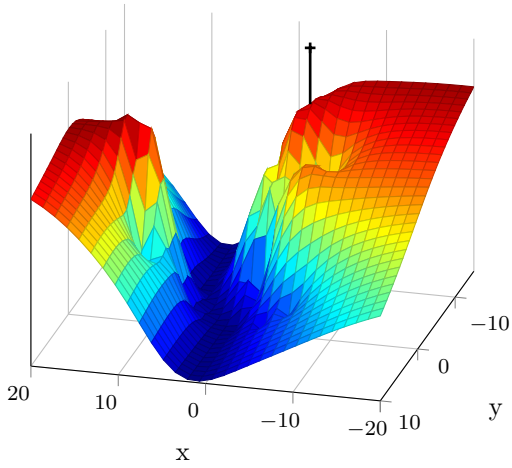
larger (positive) values of x and a large log-likelihood corresponds to infeasible values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$. Panel 7.4c shows the concentrated log-likelihood with the addition of the smooth-box prior. The area corresponding to infeasible values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ is penalized, and the maximum closer to the true position is emphasized.

Table 7.5 compares the results of using different variants of concentrated log-likelihood maximization for position estimation using the real data collected in our experiments. The statistics presented correspond to position estimate errors for all the nodes and datasets from our three experiments. The solver used in all cases is again the L-BFGS-B constrained to $0 < \alpha < 1$ and $l_0 > 0$, and jointly optimizing over $\vec{\boldsymbol{s}}$, α and l_0. The initial values for α and l_0 were 0.5 and 1 respectively. Those for $\vec{\boldsymbol{s}}$ were calculated keeping $\alpha = 0.5$ and $l_0 = 1$ fixed and using a grid-search over $\vec{\boldsymbol{s}}$ with a grid resolution of 0.1 m to ensure discarding local maxima.

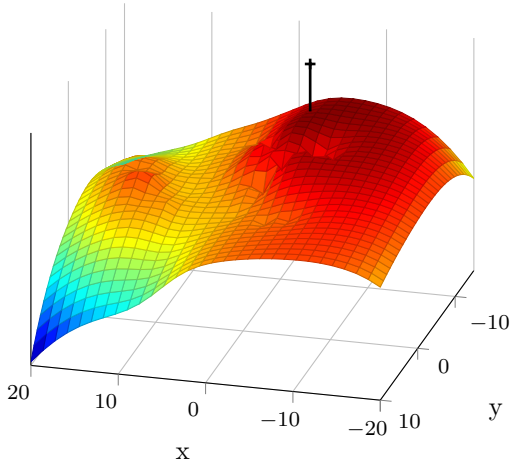
The most significant difference between using and not using priors is the maximum error, which decreases notably when using priors. These differences are due to the large errors incurred in cases in which the solver evolved towards positions corresponding to infeasible values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ when no regularization is used. This is partially solved by the regularization. Using the smooth-box prior leads to slightly smaller errors than when using Gaussians. The reason is that the smooth-box has well-differentiated areas corresponding to unaffected and penalized values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ separated by a clear edge (see Figure 7.2), whereas the Gaussian is smoother and affects all positions whose associated $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ is not exactly equal to the prior



(a) $L(\boldsymbol{z}; \vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}}, \boldsymbol{\theta}_{\mathrm{c}})$.

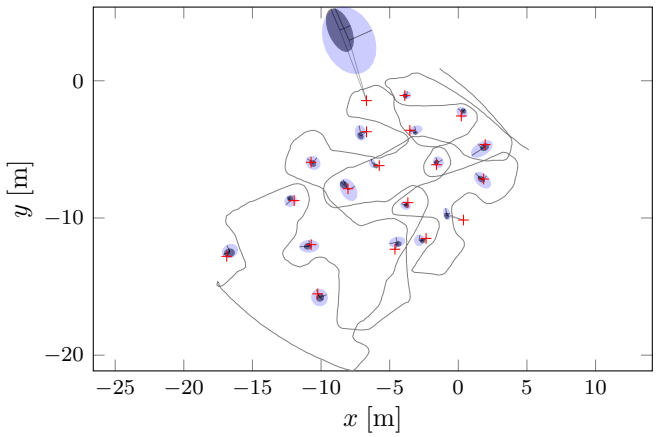


(b) $L_{mc}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \alpha, l_0)$.

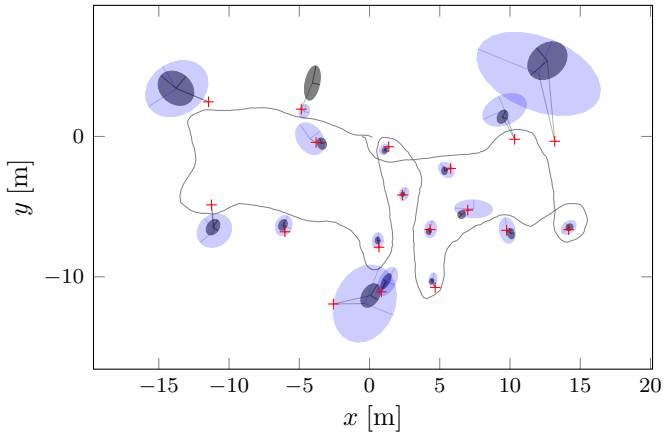


(c) $L_{mc}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \alpha, l_0) + L_p(\hat{\boldsymbol{\theta}}_{\mathrm{m}})$

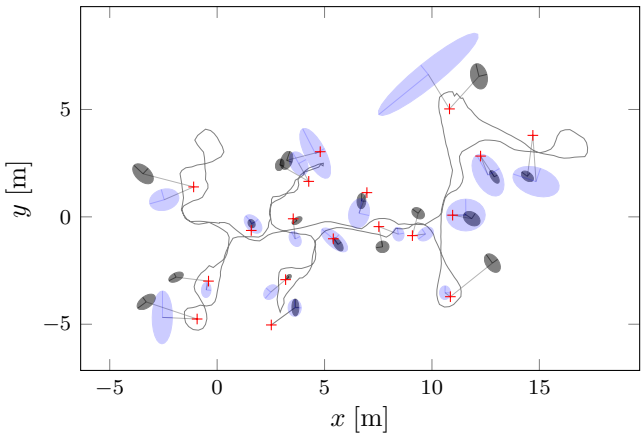
Figure 7.4: Effects of the smooth-box prior on $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ ($\alpha = 0.5$ and $l_0 = 1$).



(a) Basketball field.

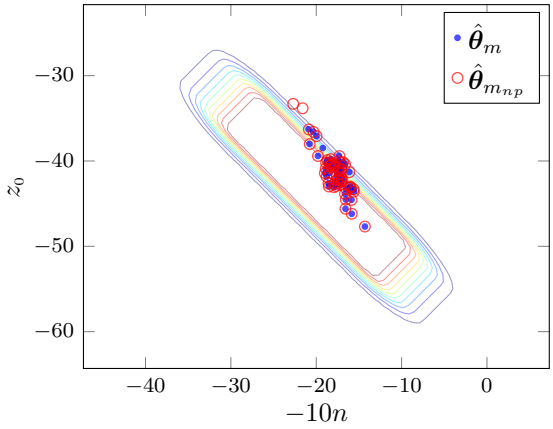


(b) Lobby.

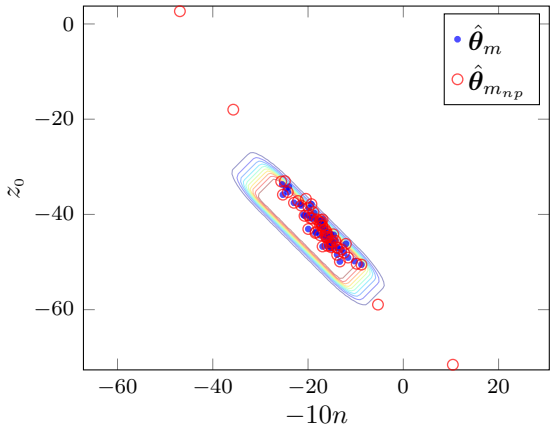


(c) Office.

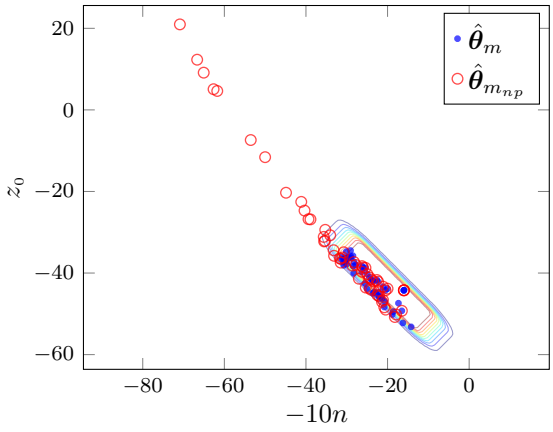
Figure 7.5: Example of localization results with their 95% confidence error ellipses in the three environments. True positions are marked with a '+' symbol, and the robot trajectories with solid curves. Results marked in light blue correspond to using the model (6.17) with correlated noise, while those in dark grey correspond to assuming IID noise ($\mathbf{\Sigma} = \sigma^2\mathbf{I}$).



(a) Basketball field



(b) Lobby



(c) Office

Figure 7.6: Estimates $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ assuming correlated noise when using smooth-box and no priors ($\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{m_{np}}}$ respectively).

mean (with zero gradient). This renders the smooth-box prior more effective for confining the model parameters within a desired area. The last row in Table 7.5 presents the error statistics associated with ignoring the correlation (IID measurements). As we can see, this increases the error in all environments.

The effect of the smooth-box priors on the estimates of $\boldsymbol{\theta}_{\mathrm{m}}$ can be seen clearly in the panels 7.6a, 7.6b and 7.6c of Figure 7.6. The panels present the values of $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ associated with position estimates calculated with and without using smooth-box priors. In the basketball field, the usage of priors is not so critical as in the other environments, as the values of $\hat{\boldsymbol{\theta}}_{\mathrm{m_{np}}}$ still fall within the feasibility region. In the lobby and office, however, there are clear cases in which $z_0 > 0$ and $n < 0$. Using the smooth-box priors corrected this behaviour, keeping $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ within the feasibility region delimited by the prior in all environments.

Finally, panels 7.5a, 7.5b and 7.5c compare position estimates together with their covariances (presented as error ellipses) under the IID (dark grey) and autocorrelated (light blue) noise assumptions. The error ellipses are calculated using the Hessian of the concentrated log-likelihoods with respect to $\vec{\boldsymbol{s}}$. These figures clearly show how ignoring the autocorrelation can lead to overconfident estimates when using a large number of spatially close measurements.

7.5 Concentrated Log-likelihood With Respect to the Position

In the previous section we eliminated the dependency of the position estimator on the model parameters $\boldsymbol{\theta}_{\mathrm{m}}$ and σ by partially optimizing the log-likelihood function over them. This was possible because the partial optimization led to analytical estimators $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ and $\widehat{\sigma^2}$ which depend only on $\vec{\boldsymbol{s}}$ and the remaining model parameters (α and l_0). Substituting them for $\boldsymbol{\theta}_{\mathrm{m}}$ and σ^2 respectively in the log-likelihood resulted in a concentrated log-likelihood whose dependency on $\boldsymbol{\theta}_{\mathrm{m}}$ and σ^2 has been eliminated. After all the analytical artillery had been used, we had to rely on a numerical solver to find the maximum of the concentrated log-likelihood and, henceforth, the position estimate $\hat{\vec{\boldsymbol{s}}}$.

In this section we follow a similar principle in that the aim is to reduce the dimensionality of the log-likelihood input space. However, instead of eliminating the dependency on $\boldsymbol{\theta}_{\mathrm{m}}$ and σ, our goal is to eliminate the dependency on $\vec{\boldsymbol{s}}$, leading to an non-linear optimization over the model parameters only. Once estimated, these parameters will be used to estimate the position. The difference is that, unlike when concentrating with respect to $\boldsymbol{\theta}_{\mathrm{m}}$ and σ, now we cannot find an analytical estimator $\hat{\vec{\boldsymbol{s}}}$ by partially optimizing the log-likelihood. The idea is to use instead the linear estimators proposed in Section 4.5.

Recall from Section 4.5 and Subsection 5.3.3 that, in general, we obtained

quite good results using GLS position estimator combined with ML distance estimators, which we referred to as the GLS-ML algorithm. Recall also from (4.100) that the GLS position estimator is

$$\hat{\vec{\boldsymbol{s}}} = [\boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \boldsymbol{A}_{\mathbf{L}}]^{-1} \boldsymbol{A}_{\mathbf{L}}^t \hat{\boldsymbol{\Sigma}}_{\mathbf{L}}^{-1} \hat{\boldsymbol{b}}_{\mathrm{L}}, \tag{7.25}$$

where $\boldsymbol{A}_{\mathbf{L}}$ is defined in (4.84), $\hat{\boldsymbol{b}}_{\mathrm{L}}$ in (4.92) and $\hat{\boldsymbol{\Sigma}}_{\mathbf{L}}$ for the ML distance estimators is given in (4.94), which is calculated using estimates of squared distances as explained in Subsection 4.5.2. This estimator depends only on $\boldsymbol{\theta}_{\mathrm{m}}$ and σ. Substituting $\hat{\vec{\boldsymbol{s}}}$ for $\vec{\boldsymbol{s}}$ in (7.9) we get the following expression of the log-likelihood:

$$\begin{aligned} L_s(\boldsymbol{z};\boldsymbol{\theta}) &\equiv L(\boldsymbol{z};\hat{\vec{\boldsymbol{s}}},\boldsymbol{\theta}_{\mathrm{m}},\boldsymbol{\theta}_{\mathrm{c}}) \\ &= -\frac{1}{2}\ln\left(|\boldsymbol{\Sigma}|\right) - \frac{1}{2}J_s - \frac{K}{2}\ln(2\pi), \end{aligned} \tag{7.26}$$

where

$$J_s(\boldsymbol{\theta}) \equiv \left(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}}\right)^t \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{z} - \boldsymbol{A}\boldsymbol{\theta}_{\mathrm{m}}\right). \tag{7.27}$$

Note the difference between the design matrices $\boldsymbol{A}$ and $\boldsymbol{A}_{\mathbf{L}}$. The first one is from the RSS model itself, whereas the latter one is from the linear position estimator (see (4.84)). More importantly, $\boldsymbol{A}_{\mathbf{L}}$ does not depend on the position $\vec{\boldsymbol{s}}$, whereas $\boldsymbol{A}$ does. However, because of the substitution of $\hat{\vec{\boldsymbol{s}}}$ for $\vec{\boldsymbol{s}}$ in the log-likelihood, this dependency is transformed into a dependency on $\boldsymbol{\theta}_{\mathrm{m}}$ and σ. Therefore (7.27) does not depend on $\vec{\boldsymbol{s}}$, but only on the model parameters, $\boldsymbol{\theta} = (\boldsymbol{\theta}_m^t, \boldsymbol{\theta}_c^t)^t$. The result is a log-likelihood whose dependency on the position has been eliminated. Following the notation used throughout this chapter, we can say that the log-likelihood has been concentrated with respect to the position.

In order to estimate the model parameters, we find the value of $\boldsymbol{\theta}$ that maximizes the concentrated log-likelihood. Similarly as in the previous section, we use regularization to prevent the convergence issues, resulting in a MAP estimator. This time the priors directly affect $\boldsymbol{\theta}_{\mathrm{m}}$, which we optimize over. The MAP estimator of the model parameters is then:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[L_s(\boldsymbol{z};\boldsymbol{\theta}_{\mathrm{m}},\boldsymbol{\theta}_s) + L_p(\boldsymbol{\theta}_{\mathrm{m}})\right], \tag{7.28}$$

where note that we again use priors only on $\boldsymbol{\theta}_{\mathrm{m}}$. Once $\hat{\boldsymbol{\theta}}$ has been calculated we can use it to calculate $\hat{\vec{\boldsymbol{s}}}$ as in (7.25).

The complete algorithm that we use to estimate position exploiting these ideas is presented schematically in Table 7.6. The algorithm accepts as inputs the observations $\boldsymbol{z}$, the trajectory of the robot $\boldsymbol{R}$ and a matrix $\boldsymbol{G}_{\boldsymbol{\theta}_{\mathrm{m}}}$ containing values of $\boldsymbol{\theta}_{\mathrm{m}}$ arranged in a grid. The first two lines define the functions that

Table 7.6: Localization algorithm using the log-likelihood concentrated with respect to $\vec{\boldsymbol{s}}$.

1	$[\hat{\vec{\boldsymbol{s}}}, \mathrm{Cov}\,[\vec{\boldsymbol{s}}]]$=loc_cltheta($\boldsymbol{z}$, $\boldsymbol{R}$, $\boldsymbol{G}_{\boldsymbol{\theta}_\mathrm{m}}$)
2	$L_{sp}(\boldsymbol{\theta}_\mathrm{m}, \boldsymbol{\theta}_\mathrm{c}) \equiv L_s(\boldsymbol{z}; \boldsymbol{\theta}_\mathrm{m}, \boldsymbol{\theta}_\mathrm{c}) + L_p(\boldsymbol{\theta}_\mathrm{m})$
3	$L_{mp}(\vec{\boldsymbol{s}}, \alpha, l_0) \equiv L_{mc}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \alpha, l_0) + L_p(\hat{\boldsymbol{\theta}}_\mathrm{m})$
4	$\hat{\boldsymbol{\theta}}_{\mathrm{c}0} = (2\sqrt{2}, 0.5, 1)^t$
5	$\hat{\boldsymbol{\theta}}_{\mathrm{m}0}$=grid_search($L_{sp}$, $\boldsymbol{G}_{\boldsymbol{\theta}_\mathrm{m}}$, $\hat{\boldsymbol{\theta}}_{\mathrm{c}0}$)
6	$[\hat{\boldsymbol{\theta}}_{\mathrm{m}1}, \hat{\boldsymbol{\theta}}_{\mathrm{c}1}]$=maximize($L_{sp}$, $\hat{\boldsymbol{\theta}}_{\mathrm{m}0}$, $\hat{\boldsymbol{\theta}}_{\mathrm{c}0}$)
7	$\hat{\vec{\boldsymbol{s}}}_0$=lls_glsml($\hat{\boldsymbol{\theta}}_{\mathrm{m}1}$, $\hat{\sigma}_{\mathrm{c}1}$)
8	$[\hat{\vec{\boldsymbol{s}}}, \hat{\alpha}, \hat{l}_0]$=maximize($L_{mp}$, $\hat{\vec{\boldsymbol{s}}}_0$, $\hat{\alpha}_{\mathrm{c}1}$, $\hat{l}_{0\mathrm{c}1}$)
9	$\mathrm{Cov}\,[\vec{\boldsymbol{s}}] = -\boldsymbol{H}_{\vec{\boldsymbol{s}}}^{-1}(L_{mp})\big\vert_{\hat{\vec{\boldsymbol{s}}}, \hat{\alpha}, \hat{l}_0}$

will be maximized in the algorithm. Line 2 defines L_{sp} as the concentrated log-likelihood with respect to $\vec{\boldsymbol{s}}$ with the addition of the log-prior. Line 3 defines the L_{mp} as the concentrated log-likelihood with respect to $\boldsymbol{\theta}_\mathrm{m}$ plus the log-prior $L_p(\hat{\boldsymbol{\theta}}_\mathrm{m})$. Therefore, it is the function to be maximized to estimate $\vec{\boldsymbol{s}}$ as proposed in Section 7.4.

The maximization corresponding to (7.28) is calculated in Line 6. As usual, we use the L-BFGS-B solver with the bounding restrictions of $0 < \alpha < 1$ and $l_0 > 0$. The solver requires initial values $\hat{\boldsymbol{\theta}}_{\mathrm{c}0}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{m}0}$. For the first one we will always use the same values: $\sigma = 2\sqrt{2}$, $\alpha = 0.5$ and $l_0 = 1$ (Line 4). The value $\hat{\boldsymbol{\theta}}_{\mathrm{m}0}$ is calculated in the grid-search in line 5 as the element of $\boldsymbol{G}_{\boldsymbol{\theta}_\mathrm{m}}$ that maximizes L_{sp} while keeping $\hat{\boldsymbol{\theta}}_{\mathrm{c}0}$ fixed. The grid used in all cases is the one represented using black dots in Figure 7.2a. Once that $\hat{\boldsymbol{\theta}}_{\mathrm{m}1}$ and $\hat{\boldsymbol{\theta}}_{\mathrm{c}1}$ have been calculated after the maximization of line 6, line 7 computes $\hat{\vec{\boldsymbol{s}}}_0$ using the GLS-ML linear estimator (7.25), which requires as arguments $\hat{\boldsymbol{\theta}}_\mathrm{m}$ and $\hat{\sigma}$ from $\hat{\boldsymbol{\theta}}_{\mathrm{c}1}$ (denoted as $\hat{\sigma}_{\mathrm{c}1}$ in the listing).

Unlike in the previous section, eliminating the dependency of the log-likelihood on $\vec{\boldsymbol{s}}$ has not been done through an exact analytical partial maximization, but simply substituting the linear estimator $\hat{\vec{\boldsymbol{s}}}$ for $\vec{\boldsymbol{s}}$. Therefore, although the estimate $\hat{\vec{\boldsymbol{s}}}$ corresponds to a maximum of the concentrated and regularized log-likelihood as defined in this section, it does not necessarily correspond to a maximum of the original log-likelihood. To complete the localization process we do a final maximization of the concentrated log-likelihood with respect to $\boldsymbol{\theta}_\mathrm{m}$ and σ with regularization as proposed in Section 7.4. That is:

$$\begin{pmatrix} \hat{\vec{\boldsymbol{s}}} & \hat{\alpha} & \hat{l}_0 \end{pmatrix}^t = \underset{\vec{\boldsymbol{s}}, \alpha, l_0}{\mathrm{argmax}} \left[L_{mc}(\boldsymbol{z}; \vec{\boldsymbol{s}}, \alpha, l_0) + L_p(\hat{\boldsymbol{\theta}}_\mathrm{m}) \right], \tag{7.29}$$

Table 7.7: Statistics of localization errors for the algorithm in Table 7.6.

Prior	Environment	min	mean	median	max
Line 7	**Basketball field**	0.09	0.57	0.53	2.36
	Lobby	0.06	1.52	0.59	7.72
	Office	0.03	1.22	0.92	2.91
Line 8	**Basketball field**	0.02	0.44	0.29	4.61
	Lobby	0.02	1.04	0.42	6.68
	Office	0.07	0.85	0.74	2.16

where the initial values for the solver are $\hat{\vec{\boldsymbol{s}}}_0$ from line 7 and $\hat{\alpha}_{\mathrm{c1}}$ and $\hat{l}_{0\mathrm{c1}}$ from $\hat{\boldsymbol{\theta}}_{\mathrm{c1}}$ calculated in line 6. Finally, line 9 calculates the covariance of the position estimate using the Hessian of L_{mp} with respect to the position evaluated at the estimates.

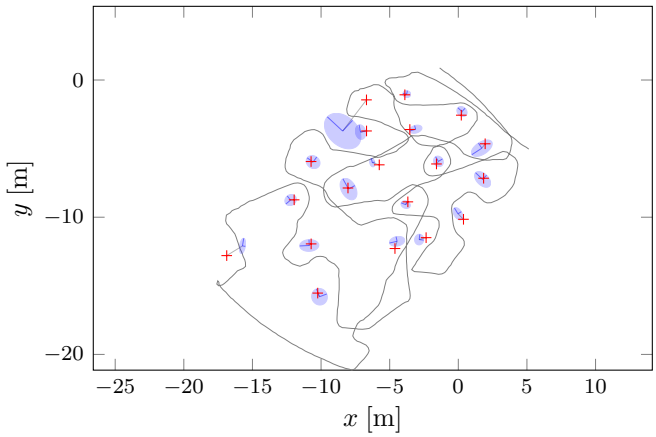
Table 7.7 presents the error statistics of the position estimates using the algorithm in Table 7.6 for the different environments as calculated in lines 7 and 8. In all cases, the prior used was the smooth-box prior described in the previous section.[9] As we can see, using the concentrated log-likelihood with respect to $\vec{\boldsymbol{s}}$ already gives a reasonably small average error. The step in line 8 is then a refinement that further reduces the error to values very similar to those obtained by maximizing the concentrated log-likelihood with respect to $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ (see Table 7.5).

The refinement step of Line 8 has the additional advantage of resulting in a better covariance of the position estimates. Effectively, one could use the covariance of the GLS-ML linear position estimator as calculated in Section 4.5. Note, however, that the estimator in line 7 uses an RSS model that ignores the autocorrelation among measurements. Maximizing the concentrated log-likelihood as in line 8 takes back this correlation into account, and results in more realistic estimator covariances.[10] Examples of the final localization results in our different experimental environments can be seen in Figure 7.7.

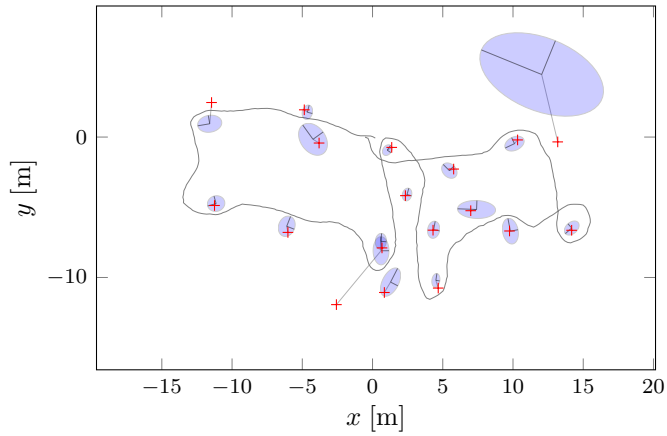
Using the concentrated log-likelihood function presented in this section has certain advantages compared to using the concentrated log-likelihood with respect to $\boldsymbol{\theta}_m$ and σ as explained in the previous section. As argued in Section 7.3, in some cases it is easier to have an initial guess at the model parameters than at the position of the target node. Intuitively, model parameters represent types of environments, whereas positions are entirely specific to each deployment and cannot be generalized in the same way. Note that in the previous section, we

[9] We also tried to use Gaussian priors, but they were unable to properly confine the model parameters within the region of feasible values.

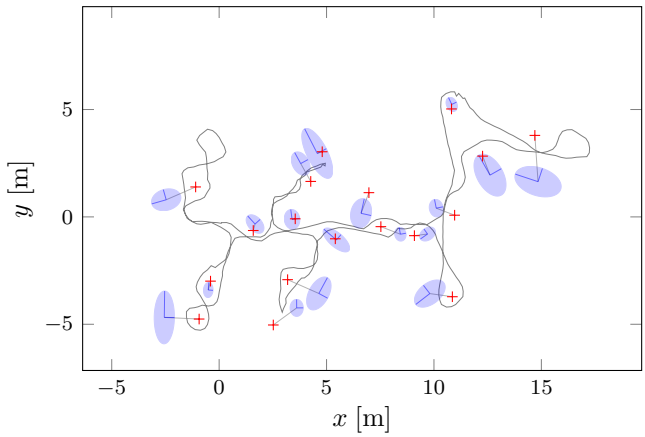
[10] This is judged visually.



(a) Basketball field.



(b) Lobby.



(c) Office.

Figure 7.7: ML localization results when using the concentrated log-likelihood with respect to $\vec{s}$

had to use one grid of initial position estimates in each environment for the grid-search, whereas in this section we have used the same grid of initial values for $\boldsymbol{\theta}_{\mathrm{m}}$ for all the environments.

Furthermore, in our case the grids used for the initial position estimates had a 0.1 m resolution, which resulted in approximately 200×200 test points. In contrast, the grid in Figure 7.2a contains 7×10 points, which represents a significant reduction in the number of test points. Depending on the particular setup, this could lead to significant computational savings. One should notice, however, that evaluating the concentrated log-likelihood with respect to the position requires inverting two matrices, $\boldsymbol{\Sigma} \in \mathbb{R}^{K \times K}$ and $\boldsymbol{\Sigma}_{\mathbf{L}} \in \mathbb{R}^{(K-1)\times(K-1)}$, whereas evaluating the one concentrated with respect to the model parameters requires the inversion of only $\boldsymbol{\Sigma}$.[11]

All in all, using the concentrated log-likelihood with respect to the position is yet another method to be added to the list of possible resources to solve localization problems. It can also be seen as an example of the potential of the closed form linear position estimators proposed in Section 4.5 for inspiring new lines of research, especially the family of GLS algorithms. Being analytical close form solutions that do not require an initial position estimate, they can be incorporated as position estimators into other more involved algorithms. This can lead to e.g. computational savings, simplifications and even new algorithms that otherwise could not be directly developed, such as the one presented in this section. In addition, because their closed form, one can manipulate them algebraically and compute their derivatives, enabling further analytical studies also in closed form. This is all possible thanks to their demonstrated good performance, which, under the appropriate experimental conditions, renders the GLS algorithms into reasonable approximations of the ML estimator.

7.6 Conclusions

This chapter was dedicated to the problem of position estimation in cases when the model parameters are not available *a priori*. Our approach is based on methods that maximize the log-likelihood of the measurements jointly over the position and model parameters. This increases the risk of the appearance of problems related to identifiability and local maxima.

Three methods have been proposed: one which maximizes the log-likelihood following sequential orthogonal steps (LMIR) and that reduce the dimension of the input space through a concentration of the log-likelihood function.

LMIR was tested with several types of polynomial models and using MoGs to

[11] In practice, as usual we avoid to invert any matrix explicitly; the aim of the text is to simply provide a comparison of the cost between the evaluation of the two concentrated log-likelihoods.

explain the residuals, all assuming IID observations. Because the same dataset is used for the localization and model identification, the risk of model over-fitting is not as relevant, unlike the case of simple localization. The model specialization (usage of higher degree polynomial and MoGs) now increased the accuracy of the estimates compared to using the traditional log-normal model.

The methods exploiting concentrated log-likelihoods all considered spatially autocorrelated measurements, which solve the over-confidence problem discussed in previous chapters. Concentrating the log-likelihood with respect to $\boldsymbol{\theta}_m$ and σ was done through a partial analytical optimization over them. Therefore, essentially the log-likelihood to be maximized does not change, but only the dimensionality of its input space. A second reading of this is that concentrating the log-likelihood does not eliminate the identifiability and convergence issues, should they be present. It has been observed experimentally that the approach can be very sensitive to the initial values of the position given to the solver and the geometrical configuration of the problem.[12]

The problems that we observed could be diagnosed by observing the estimates $\hat{\boldsymbol{\theta}}_m$ after convergence. These can obtain infeasible values leading to large localization errors and position estimates' covariance matrices. To overcome this problem we propose using regularization penalizing the estimator function $\hat{\boldsymbol{\theta}}_m$, which is equivalent to MAP estimation with a prior on $\hat{\boldsymbol{\theta}}_m$. In essence, this penalizes positions that can result in infeasible model parameters. This penalization is achieved trough the dependency of $\hat{\boldsymbol{\theta}}_m$ on the position over which the maximization takes place. The advantage of this scheme is that it allows using the concentrated log-likelihood with respect to $\hat{\boldsymbol{\theta}}_m$ and σ while not being explicitly dependent on them, and thus retaining the low dimensionality of the concentrated log-likelihood. Using our experimental setup we have demonstrated the effectiveness of the proposed technique, which reduced significantly the error in cases affected by identifiability and convergence issues.

The last method explored concentrates the log-likelihood with respect to the position relying on the GLS-ML linear position estimator proposed in Section 4.5, and not on a partial analytical optimization of the log-likelihood over $\vec{\boldsymbol{s}}$. Thus, the maximum of this concentrated log-likelihood does not necessarily coincide with the maximum of the original log-likelihood. Nevertheless, it results in reasonable position estimates, which can then be used as initial values for other methods. The main advantage of this method is that it does not need initial position estimates, only model parameters.

[12] We suspect that the origin of this issue is that certain problem geometries can lead to a problem with a non-identifiable scale. This will be investigated in future research.

Chapter 8

Summary and Conclusions

The work presented in this thesis provides practical solutions for the localization problem in real *ad-hoc* WSN deployments using RSS measurements and beacons or reference devices. The motivational type of target application is an emergency and rescue operation, typically characterized by a lack of pre-installed infrastructure (e.g. reference devices) and the impossibility of collecting on-site training data. These are two real challenges that practical localization methods must address to be eligible for these scenarios.

We propose using a SLAM-capable robot as a mobile beacon to solve the problem of a lack of infrastructure. This brings with it additional important advantages. First, the robot creates a map of the environment, which can be invaluable for the personnel involved in rescue operations. Second, it makes it possible to collect large amounts of spatially rich data. For a researcher, this opens up the possibility of studying in more detail the spatial behaviour of the signals exploited for localization, which in turn makes it possible to create better models and position estimation methods.

The challenge of lacking training data implies that the localization algorithms must be able to work in diverse environments without *a priori* knowledge of the radio propagation conditions. Along the journey of this thesis, different localization methods have been proposed and tested. These are not necessarily applicable in all possible conditions encountered in real emergency scenarios. In this sense, the author envisions their applicability in emergency and rescue scenarios as the steering agent of the research, and the scientific contributions presented have arisen from the work done towards achieving this ultimate goal.

The work began with an extensive data collection campaign in which we retrieved measurements from three different environments using a robot and a WSN. These data have been used to characterize the RSS models and to validate

and compare the positioning methods under study.

The estimation of the nodes' position is posed as a source localization problem in which the position of each node is calculated independently using the data collected by the respective node. The localization algorithms analyzed use simple parametric RSS-distance models, among which the classical log-normal model has played a fundamental role. Using our experimental data, we have shown how much the model parameters can vary as a consequence of the HW differences among nodes and the environmental inhomogeneity. Because of this variability, using one model to explain the observations of each node results in better RSS predictions than using a single site-specific model for all of them. We have denoted these two paradigms as using individual (IM) versus common models (CM).

Although HW and environmental factors affect each of the model parameters differently, these factors interact in a complex manner, and it is not possible to evaluate accurately their effects independently. The linearity restriction of the log-normal model additionally induces correlation between the path-loss exponent and the reference power, making this interaction more complex. As a result of the observed variability and the effect of these interactions, we propose to characterize real deployments using joint PDFs of the model parameters instead of independent deterministic values. This can be used, for example, to generate more realistic simulations of RSS measurements from WSN deployments. The mean, variance and correlation values given in our studies can be used as examples of the statistics of these distributions.

In order to evaluate the impact of the variability in the model parameters in localization algorithms, we presented a sensitivity study of ML localization. This sensitivity has been shown to be relevant and deserves attention. Using a CM identified with training data collected by an increasing number of nodes does not automatically imply a reduction of the mean localization error. The result is simply a more robust model against the effects of the parameters' variability. Using IMs is notably more effective, and it resulted in decreases of the mean error of up to 65% in our experiments. This relative increase of the performance is more relevant in environments where the log-normal model is a better predictor.

Besides ML, other localization methods were also proposed and studied using LS estimation. Two new families of distance-based linear position estimators have been devised based on an existing OLS solution: GLS and IGLS. These estimators have the advantage of not requiring an initial position estimate, as opposed to e.g. ML. The performance of the algorithms has been studied and compared using simulations of observations under the log-normal and AGN models, being the second one applicable to time-based distance estimation. The algorithms under the log-normal model were also tested with real data. The GLS family has a closed form solution, and therefore it is very computationally efficient.

It demonstrated an excellent performance for its computational requirements, with average errors of 0.71, 2.14 and 1.3 m in our environments when used with the ML distance estimator. The ML variant of the IGLS family remarkably outperformed ML in some of our experiments, showing sub-metre average errors in all our environments.

Because of its desirable asymptotic properties and the better general performance under bad geometrical conditions, ML was still chosen as the central position estimation method. Using a robot as a mobile beacon with high sampling rates creates two characteristics problems for ML localization: first the appearance of a characteristic type of local maxima, and second, the overconfidence of estimates under the IID errors assumption.

The first problem can be solved e.g. starting a gradient based solver with a grid-search (i.e. brute force), which is well known to be computationally expensive. As an alternative, we devised a new method aiming at a more efficient search. In essence, it complements gradient-based solvers with a heuristic that tries to determine whether the maximum reached at convergence is local or not by exploring the neighbourhood guided by the shape of the log-likelihood function. In case the maximum is found to be local, the gradient solver continues its iterations towards another higher maxima. We called this algorithm Maximize-Explore.

The over-confidence problem was addressed using three paths: a) decreasing manually the amount of information provided by each sample. This requires manually tuning the parameter that controls the degree of information being lost, which in turn is data-size dependent. In addition, it ignores potential effects due to high local spatial concentration of samples, which can be an issue when the observations are spatially correlated. b) Discarding correlated measurements by selecting data only from control points distant enough from one another. The advantage is a reduction of the datasize, with its consequent computational savings. The disadvantage is a loss of information as more data are dropped, which decreases the accuracy. The proposed sample selection algorithm still requires manually tuning the minimum distance among control points. c) Explicitly modelling the spatial autocorrelation. This last option was the preferred one, as it retains a good level of accuracy while enables the automatic selection of the parameters involved using joint localization and model identification techniques.

The incapability of the log-normal model to explain the observations in obstructed environments causes spatial autocorrelation in the residuals. In order to incorporate spatial effects into the model we added a zero-mean random Gaussian process with exponentially decaying autocorrelation to the error term. As a result, two independent random variables take into account the model uncertainty: one autocorrelated modelling the shadow fading and another one IID representing any other source of uncertainty. The empirical correlograms showed a good adequacy of the model in the basketball field and lobby, and arguable in

the office. The aggregate trend, however, was still a decaying exponential, and for convenience we used the same type of model in all the environments.

Finally, in the last chapter we addressed the localization problem with no prior deterministic knowledge of the model parameters and under spatially autocorrelated noise. Three methods were proposed based on a joint estimation of position and model parameters. The first method maximizes the log-likelihood in orthogonal steps in the position's and model parameters' subspaces, respectively, starting from an initial guess at the model parameters. The second method exploits the principle of separability of LS to concentrate the log-likelihood with respect to the parameters affecting the mean, $\boldsymbol{\theta}_m$, and the total variance, σ. This results in a log-likelihood dependent on the position and two other model parameters. The third method substitutes the position in the log-likelihood by the GLS-ML linear estimator proposed and studied previously in this thesis. As a result, the dependency on the position is eliminated, and the optimization can be done entirely in the model parameters' subspace. Compared with concentrating the log-likelihood with respect to $\boldsymbol{\theta}_m$ and σ, this last method required a grid with significantly fewer points, resulting in a smaller number of log-likelihood evaluations. It also serves to demonstrate the possibilities that the newly proposed linear estimators offer for creating novel types of algorithms thanks to their closed analytical form, their simplicity and their good performance.

Due to the increase in the number of free explanatory variables, a joint optimization of the log-likelihood using either of the methods proved to be sensitive to the initial value given to the solver and the problem geometry, which occasionally led to large localization errors. When the log-likelihood was concentrated with respect to $\boldsymbol{\theta}_m$ and σ, in cases where the problem occurred it could be diagnosed by monitoring the value of the expression $\hat{\boldsymbol{\theta}}_m$; it obtains infeasible values with negative path-loss exponents and positive reference powers. As a solution we added a regularization term on $\hat{\boldsymbol{\theta}}_m$, leading to a concentrated log-likelihood that maintains the low dimensionality of the input space. The effect of this addition is a penalization of positions associated with infeasible estimates of the model parameters. Using the appropriate priors, this technique guides gradient-based solvers towards a possitions associated to feasible model parameter estimates. The approach proved to be effective, and allowed us to reach sub-metre average accuracy in all our experimental setups using a prior especially designed to confine the model parameters in a desired feasibility area.

To conclude, the studies and methods proposed show how one can exploit the rich spatial information provided by mobile robots to estimate the nodes' position accurately, precisely and efficiently in real *ad-hoc* deployments without *a priori* knowledge of the environmental characteristics. The results demonstrate the great potential for using mobile robots as assistive localization devices, especially in WSNs.

References

[Agrawal and Patwari, 2009] Agrawal, P. and Patwari, N. (2009). Correlated link shadow fading in multi-hop wireless networks. *IEEE Transactions on Wireless Communications*, 8(8):4024–4036.

[Agrawal and Patwari, 2011] Agrawal, P. and Patwari, N. (2011). Kernel methods for RSS-based indoor localization. In Zekavat, R. and Buehrer, R., editors, *Handbook of Position Location: Theory, Practice, and Advances*, chapter 14. John Wiley & Sons, Inc.

[Akerberg, 1989] Akerberg, D. (1989, 1-3 May). Properties of a TDMA pico cellular office communication system. In *IEEE 39th Vehicular Technology Conference*, volume 1, pages 186–191, San Francisco, CA, USA.

[Al-Dhalaan and Lambadaris, 2010] Al-Dhalaan, A. and Lambadaris, I. (2010, 23-27 May). Wireless sensor network localization with spatially correlated shadowing. In *IEEE International Conference on Communications (ICC)*, pages 1–6, Cape Town, South Africa.

[Algans et al., 2002] Algans, A., Pedersen, K., and Mogensen, P. (2002). Experimental analysis of the joint statistical properties of azimuth spread, delay spread, and shadow fading. *IEEE Journal on Selected Areas in Communications*, 20(3):523–531.

[Allsebrook and Parsons, 1977] Allsebrook, K. and Parsons, J. (1977). Mobile radio propagation in British cities at frequencies in the VHF and UHF bands. *Proceedings of the Institution of Electrical Engineers*, 124(2):95–102.

[Amemiya, 1985] Amemiya, T. (1985). *Advanced econometrics*. Blackwell, Oxford.

[Amitay, 1992] Amitay, N. (1992). Modeling and computer simulation of wave propagation in lineal line-of-sight microcells. *IEEE Transactions onVehicular Technology*, 41(4):337–342.

[Arbia, 2010] Arbia, G. (2010). *Spatial Econometrics: Statistical Foundations and Applications to Regional Convergence*. Advances in Spatial Science. Springer.

[Bertoni, 1999] Bertoni, H. L. (1999). *Radio Propagation for Modern Wireless Systems*. Prentice Hall Professional Technical Reference.

[Biswas and Ye, 2004] Biswas, P. and Ye, Y. (2004, 26-27 Apr.). Semidefinite programming for ad hoc wireless sensor network localization. In *Third International Symposium on Information Processing in Sensor Networks*, pages 46–54, Berkeley, CA, USA.

[Bjorkbom et al., 2013] Bjorkbom, M., Timonen, J., Yigitler, H., Kaltiokallio, O., Vallet García, J. M., Myrsky, M., Saarinen, J., Korkalainen, M., Cuhac, C., Jantti, R., Virrankoski, R., Vankka, J., and Koivo, H. (2013). Localization services for online common operational picture and situation awareness. *IEEE Access*, 1:742–757.

[Botev et al., 2010] Botev, Z. I., Grotowski, J. F., and Kroese, D. P. (2010). Kernel density estimation via diffusion. *Annals of Statistics*, 38:2916–2957.

[Byrd et al., 1995] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16(5):1190–1208.

[Caballero et al., 2008a] Caballero, F., Merino, L., Gil, P., Maza, I., and Ollero, A. (2008a). A probabilistic framework for entire WSN localization using a mobile robot. *Robotics and Autonomous Systems*, 56(10):798–806.

[Caballero et al., 2008b] Caballero, F., Merino, L., Maza, I., and Ollero, A. (2008b, 19-23 May). A particle filtering method for wireless sensor network localization with an aerial robot beacon. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 596–601, Pasadena, CA, USA.

[Caballero et al., 2010] Caballero, F., Merino, L., and Ollero, A. (2010, 3-7 May). A general gaussian-mixture approach for range-only mapping using multiple hypotheses. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 4404–4409, Anchorage, AK, USA.

[Caffery, 2000] Caffery, Jr., J. J. (2000, 24-28 Sept.). A new approach to the geometry of TOA location. In *Proc. IEEE 52nd Vehicular Technology Conf. (VTS-Fall)*, volume 4, pages 1943–1949, Boston, MA, USA.

[Carbonetto, 2014] Carbonetto, P. (2014). lbfgsb-matlab project. Online: `https://github.com/pcarbo/lbfgsb-matlab`. Accessed: Mar. 2014.

[Catrein and Mathar, 2008] Catrein, D. and Mathar, R. (2008, 7-10 Dec.). Gaussian random fields as a model for spatially correlated log-normal fading. In *Australasian Telecommunication Networks and Applications Conference*, pages 153–157, Adelaide, SA, USA.

[Cavers, 2000] Cavers, J. (2000). *Mobile Channel Characteristics*. Kluwer Academic Publishers.

[Chitte et al., 2009a] Chitte, S., Dasgupta, S., and Ding, Z. (2009a). Distance estimation from received signal strength under log-normal shadowing: Bias and variance. *IEEE Signal Processing Letters*, 16(3):216–218.

[Chitte et al., 2009b] Chitte, S., Dasgupta, S., and Ding, Z. (2009b, 17-19 Oct.). Source localization from received signal strength under log-normal shadowing: Bias and variance. In *2nd International Congress on Image and Signal Processing*, pages 1–5, Tianjin, China.

[COST 231, 1991] COST 231 (1991). Urban transmission loss models for mobile radio in the 900 and 1800 MHz bands (revision 2). Technical report, European Cooperative in Field of Science and Technical Research EURO-COST 231.

[Cressie, 1993] Cressie, N. A. C. (1993). *Statistics for Spatial Data*. Wiley, revised edition.

[Dabin et al., 2003] Dabin, J., Ni, N., Haimovich, A., Niver, E., and Grebel, H. (2003, 16-19 Nov.). The effects of antenna directivity on path loss and multipath propagation in UWB indoor wireless channels. In *IEEE Conference on Ultra Wideband Systems and Technologies*, pages 305–309, Reston, VA, USA.

[Davidson and MacKinnon, 2004] Davidson, R. and MacKinnon, J. (2004). *Econometric theory and methods*. Oxford Univ. Press, New York, USA.

[de Toledo and Turkmani, 1992] de Toledo, A. and Turkmani, A. (1992, 10-13 May). Propagation into and within buildings at 900, 1800 and 2300 MHz. In *IEEE 42nd Vehicular Technology Conference*, volume 2, pages 633–636, Denver, CO, USA.

[Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

[Devasirvatham et al., 1990] Devasirvatham, D. M. J., Banerjee, C., Krain, M., and Rappaport, D. A. (1990, 16-19 Apr.). Multi-frequency radiowave propagation measurements in the portable radio environment. In *IEEE International Conference on Communications Including Supercomm Technical Sessions*, volume 4, pages 1334–1340, Atlanta, GA, USA.

[Dong et al., 2009] Dong, F., Chen, Y., Liu, J., Ning, Q., and Piao, S. (2009, 30 Sept.). A calibration-free localization solution for handling signal strength variance. In *Proceedings of the 2nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments*, pages 79–90, Berlin, Heidelberg. Springer-Verlag.

[Durgin et al., 1998] Durgin, G., Rappaport, T., and Xu, H. (1998). Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz. *IEEE Transactions on Communications*, 46(11):1484–1496.

[Eaton et al., 2014] Eaton, J. W., Bateman, D., Hauberg, S., and Wehbring, R. (2014). *GNU Octave version 3.8.2 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform. ISBN 1441413006.

[Erceg et al., 1999] Erceg, V., Greenstein, L., Tjandra, S., Parkoff, S., Gupta, A., Kulic, B., Julius, A., and Bianchi, R. (1999). An empirically based path loss model for wireless channels in suburban environments. *IEEE Journal on Selected Areas in Communications*, 17(7):1205–1211.

[Fang, 1986] Fang, B. T. (1986). Trilateration and extension to global positioning system navigation. *Journal of Guidance, Control, and Dynamics*, 9(6):715–717.

[Fang et al., 2010] Fang, Z., Zhao, Z., Geng, D., Xuan, Y., Du, L., and Cui, X. (2010, 20-23 June). RSSI variability characterization and calibration method in wireless sensor network. In *IEEE International Conference on Information and Automation*, pages 1532–1537, Harbin, Heilongjiang, China.

[Ferris et al., 2006] Ferris, B., Haehnel, D., and Fox, D. (2006, 16-19 Aug.). Gaussian processes for signal strength-based location estimation. In *Proceedings of Robotics: Science and Systems*, pages 1–8, Philadelphia, PA, USA.

[Flåm et al., 2010] Flåm, J., Kraidy, G., and Ryan, D. (2010, 16-19 May). Using a sensor network to localize a source under spatially correlated shadowing. In *IEEE 71st Vehicular Technology Conference (VTC 2010-Spring)*, pages 1–5, Taipei, China.

[Foy, 1976] Foy, W. (1976). Position-location solutions by Taylor-series estimation. *IEEE Transactions on Aerospace and Electronic Systems*, AES-12(2):187–194.

[Friis, 1946] Friis, H. (1946). A note on a simple transmission formula. *Proceedings of the Institute of Radio Engineers*, 34(5):254–256.

[Gezici et al., 2005] Gezici, S., Tian, Z., Giannakis, G., Kobayashi, H., Molisch, A., Poor, H., and Sahinoglu, Z. (2005). Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84.

[Gholami et al., 2013] Gholami, M., Monir Vaghefi, R., and Strom, E. (2013). RSS-based sensor localization in the presence of unknown channel parameters. *IEEE Transactions on Signal Processing*, 61(15):3752–3759.

[Gholami et al., 2011] Gholami, M., Wymeersch, H., Strom, E., and Rydstrom, M. (2011). Wireless network positioning as a convex feasibility problem. *EURASIP Journal on Wireless Communications and Networking*, 2011(1):161.

[Goldsmith, 2005] Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press, New York, USA.

[Graziosi, 1999] Graziosi, F. (1999, 16-20 Jul). Analysis of outage duration for mobile radio networks with correlated shadowings. In *IEEE 49th Vehicular Technology Conference*, volume 2, pages 948–952, Houston, TX, USA.

[Greene, 2011] Greene, W. H. (2011). *Econometric Analysis*. Prentice Hall, 7$^{\text{th}}$ edition.

[Gudmundson, 1991] Gudmundson, M. (1991). Correlation model for shadow fading in mobile radio systems. *Electronics Letters*, 27(23):2145–2146.

[Gustafsson and Gunnarsson, 2005] Gustafsson, F. and Gunnarsson, F. (2005). Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22(4):41–53.

[Gustafsson and Gunnarsson, 2007] Gustafsson, F. and Gunnarsson, F. (2007, 9-12 July). Localization in sensor networks based on log range observations. In *10th International Conference on Information Fusion*, pages 1 –8, Québec, Canada.

[Gustafsson et al., 2012] Gustafsson, F., Gunnarsson, F., and Lindgren, D. (2012). Sensor models and localization algorithms for sensor networks based

on received signal strength. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):1–13.

[Har et al., 1999] Har, D., Xia, H., and Bertoni, H. L. (1999). Path-loss prediction model for microcells. *IEEE Transactions on Vehicular Technology*, 48(5):1453–1462.

[Hashemi, 1993] Hashemi, H. (1993). The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7):943–968.

[Hero and Blatt, 2005] Hero, A. and Blatt, D. (2005, 18-23 Mar.). Sensor network source localization via projection onto convex sets (POCS). In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 689–692, Philadelphia, PA, USA.

[Hightower et al., 2001] Hightower, J., Vakili, C., and Borriello, G. (2001). Design and calibration of the spoton ad-hoc location sensing system. Unpublished.

[Howard and Pahlavan, 1990] Howard, S. and Pahlavan, K. (1990). Measurement and analysis of the indoor radio channel in the frequency domain. *IEEE Transactions on Instrumentation and Measurement*, 39(5):751–755.

[Jalden et al., 2007] Jalden, N., Zetterberg, P., Ottersten, B., Hong, A., and Thoma, R. (2007, 11-15 Mar.). Correlation properties of large scale fading based on indoor measurements. In *IEEE Wireless Communications and Networking Conference*, pages 1894–1899, Hong Kong, China.

[Kay, 1993] Kay, S. M. (1993). *Fundamentals of Statistical Signal Processing*. Prentice Hall.

[Kjærgaard and Munk, 2008] Kjærgaard, M. B. and Munk, C. V. (2008, 17-21 Mar.). Hyperbolic location fingerprinting: a calibration-free solution for handling differences in signal strength. In *Proceedings of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2008)*, pages 110–116, Hong Kong, China. IEEE Computer Society.

[Klingenbrunn and Mogensen, 1999] Klingenbrunn, T. and Mogensen, P. (1999, 19-22 Sept.). Modelling cross-correlated shadowing in network simulations. In *50th IEEE Vehicular Technology Conference, VTC 1999 - Fall*, volume 3, pages 1407–1411, Amsterdam, The Netherlands.

[Kotz et al., 2003] Kotz, D., Newport, C., and Elliott, C. (2003). The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dartmouth Computer Science.

[Lee and Buehrer, 2009] Lee, J. H. and Buehrer, R. (2009, 30 Nov. - 4 Dec.). Location estimation using differential RSS with spatially correlated shadowing. In *IEEE Global Telecommunications Conference*, pages 1–6, Honolulu, Hawai, USA.

[LeSage and Pace, 2009] LeSage, J. and Pace, R. (2009). *Introduction to Spatial Econometrics*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis.

[Letchner et al., 2005] Letchner, J., Fox, D., and LaMarca, A. (2005, 9-13 July). Large-scale localization from wireless signal strength. In *Proc. of the National Conference on Artificial Intelligence (AAAI-05)*, pages 15–20, Pittsburgh, PA, USA.

[Liberti and Rappaport, 1992] Liberti, J. and Rappaport, T. (1992, 11-14 Oct.). Statistics of shadowing in indoor radio channels at 900 and 1900 MHz. In *IEEE Military Communications Conference*, volume 3, pages 1066 –1070, San Diego, CA, USA.

[Lim et al., 2006] Lim, H., Kung, L.-C., Hou, J., and Luo, H. (2006, 23-29 Apr.). Zero-configuration, robust indoor localization: Theory and experimentation. In *Proceedings of 25th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–12, Barcelona, Spain.

[Lymberopoulos et al., 2006] Lymberopoulos, D., Lindsey, Q., and Savvides, A. (2006). An empirical characterization of radio signal strength variability in 3-D IEEE 802.15.4 networks using monopole antennas. In *Wireless Sensor Networks*, volume 3868 of *Lecture Notes in Computer Science*, pages 326–341. Springer Berlin Heidelberg.

[Mailaender, 2011] Mailaender, L. (2011, 5-8 Sept.). Geolocation bounds for received signal strength (RSS) in correlated shadow fading. In *IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–6, San Francisco, CA, USA.

[Mailaender, 2012] Mailaender, L. (2012, 10-15 June). Received signal strength (RSS) location estimation with nuisance parameters in correlated shadow fading. In *IEEE International Conference on Communications*, pages 3659 –3663, Ottawa, Canada.

[Malmgren, 1997] Malmgren, G. (1997). On the performance of single frequency networks in correlated shadow fading. *Transactions on Broadcasting*, 43(2):155–165.

[Mandayam et al., 1996] Mandayam, N. B., Chen, P.-C., and Holtzman, J. (1996, 28 Apr. - 1 May). Minimum duration outage for cellular systems: a level

crossing analysis. In *IEEE 46th Vehicular Technology Conference*, volume 2, pages 879–883, Atlanta, GA, USA.

[Manolakis, 1996] Manolakis, D. (1996). Efficient solution and performance analysis of 3-D position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1239–1248.

[Marsan et al., 1990] Marsan, M., Hess, G., and Gilbert, S. (1990). Shadowing variability in an urban land mobile environment at 900 MHz. *Electronics Letters*, 26(10):646–648.

[Matheron, 1963] Matheron, G. (1962-1963). *Traité de géostatistique appliquée*. Mémoires du Bureau de Recherches Géologiques et Minières. Éditions Technip.

[Mawira, 1992] Mawira, A. (1992, 19-21 Oct.). Models for the spatial correlation functions of the (log)-normal component of the variability of VHF/UHF field strength in urban environment. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 436–440, Boston, MA, USA.

[Menegatti et al., 2009] Menegatti, E., Zanella, A., Zilli, S., Zorzi, F., and Pagello, E. (2009, 12-17 May). Range-only slam with a mobile robot and a wireless sensor networks. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 8–14, Kobe, Japan.

[Muppirisetty et al., 2015] Muppirisetty, L. S., Svensson, T., and Wymeersch, H. (2015). Spatial wireless channel prediction under location uncertainty. *IEEE Transactions on Wireless Communications*. To be published. DOI: 10.1109/TWC.2015.2481879.

[Nguyen et al., 2011] Nguyen, H. A., Forster, A., Puccinelli, D., and Giordano, S. (2011, 21-25 Mar.). Sensor node lifetime: an experimental study. In *IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 202 –207, Seattle, WA, USA.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York, 2$^{\text{nd}}$ edition.

[Ott and Plitkins, 1978] Ott, G. and Plitkins, A. (1978). Urban path-loss characteristics at 820 MHz. *IEEE Transactions on Vehicular Technology*, 27(4):189–197.

[Papoulis and Pillai, 2002] Papoulis, A. and Pillai, U. S. (2002). *Probability, Random Variables and Stochastic Processes*. McGraw-Hill Science/Engineering/Math, 4$^{\text{th}}$ edition.

[Parsons, 2000] Parsons, J. D. (2000). *The Mobile Radio Propagation Channel, 2nd Edition*. Wiley, 2nd edition.

[Pathirana et al., 2005] Pathirana, P. N., Bulusu, N., Savkin, A. V., and Jha, S. (2005). Node localization using mobile robots in delay-tolerant sensor networks. *IEEE Transactions on Mobile Computing*, 4(3):285–296.

[Patwari and Agrawal, 2008] Patwari, N. and Agrawal, P. (2008, 22-24 Apr.). Effects of correlated shadowing: Connectivity, localization, and RF tomography. In *Proc. Int. Conf. Information Processing in Sensor Networks (IPSN)*, pages 82–93.

[Patwari et al., 2007] Patwari, N., Hero, A. O., and Costa, J. A. (2007). Learning sensor location from signal strength and connectivity. In *Advances in Information Security*, volume 30, pages 57–81. Springer.

[Patwari and Hero, 2004] Patwari, N. and Hero, A.O., I. (2004, 17-21 May). Manifold learning algorithms for localization in wireless sensor networks. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 857–60.

[Rama Rao and Balachander, 2013] Rama Rao, T. and Balachander, D. (2013). RF propagation investigations at 915/2400 MHz in indoor corridor environments for wireless sensor communications. *Progress In Electromagnetics Research B*, 47:359–381.

[Rappaport, 2002] Rappaport, T. S. (2002). *Wireless Communications: Principles and Practice*. Prentice Hall, 2nd edition.

[Rasmussen and Williams, 2006] Rasmussen, C. and Williams, C. (2006). *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, USA.

[Rasmussen and Nickisch, 2010] Rasmussen, C. E. and Nickisch, H. (2010). Gaussian processes for machine learning (GPML) toolbox. *Journal of Machine Learning Research (JMLR)*, pages 3011–3015.

[Saarinen et al., 2007] Saarinen, J., Maula, A., Nissinen, R., Kukkonen, H., Suomela, J., and Halme, A. (2007, 29 – 31 Aug.). GIMnet-infrastructure for distributed control of generic intelligent machines. In *13th IASTED International Conference on Robotics and Applications and Telematics*, pages 1–6, Würzburg, Germany.

[Saarinen et al., 2011] Saarinen, J., Paanajärvi, J., and Forsman, P. (2011, 25-30 Sept.). Best-first branch and bound search method for map based localization.

In *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems 2011*, pages 59–64.

[Salman et al., 2012] Salman, N., Ghogho, M., and Kemp, A. (2012). On the joint estimation of the RSS-based location and path-loss exponent. *IEEE Wireless Communications Letters*, 1(1):34–37.

[Salo et al., 2005] Salo, J., Vuokko, L., and Vainikainen, P. (2005, 18-22 Sept.). Why is shadow fading lognormal? In *8th International Symposium on Wireless Personal Multimedia Communications*, pages 522–526, Aalborg, Denmark.

[Savvides et al., 2001] Savvides, A., Han, C.-C., and Strivastava, M. B. (2001, 16-21 July). Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pages 166–179, Rome, Italy. ACM.

[Schabenberger and Gotway, 2004] Schabenberger, O. and Gotway, C. (2004). *Statistical Methods for Spatial Data Analysis*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis.

[Seco et al., 2009] Seco, F., Jimenez, A. R., Prieto, C., Roa, J., and Koutsou, K. (2009, 26-28 Aug.). A survey of mathematical methods for indoor localization. In *Proc. IEEE Int. Symp. Intelligent Signal Processing WISP 2009*, pages 9–14, Budapest, Hungary.

[Seidel and Rappaport, 1991] Seidel, S. and Rappaport, T. (1991, 19-22 May). 900 MHz path loss measurements and prediction techniques for in-building communication system design. In *41st IEEE Vehicular Technology Conference*, pages 613–618, St. Louis, MO, USA.

[Seidel and Rappaport, 1992] Seidel, S. and Rappaport, T. (1992). 914 MHz path loss prediction models for indoor wireless communications in multifloored buildings. *IEEE Transactions on Antennas and Propagation*, 40(2):207–217.

[Sichitiu and Ramadurai, 2004] Sichitiu, M. and Ramadurai, V. (2004, 25-27 Oct.). Localization of wireless sensor networks with a mobile beacon. In *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pages 174 – 183, Fort Lauderdale, FL, USA.

[Skidmore et al., 1996] Skidmore, R., Rappaport, T., and Abbott, A. (1996, 29 Sep. - 2 Oct.). Interactive coverage region and system design simulation for wireless communication systems in multifloored indoor environments: SMT Plus. In *5th IEEE International Conference on Universal Personal Communications*, volume 2, pages 646–650, Cambridge, MA, USA.

[Smith, 2014] Smith, T. E. (2014). Notebook on spatial data analysis. Online: `http://www.seas.upenn.edu/~ese502/#notebook`. Accessed: 2015-06-30.

[Sørensen, 1998] Sørensen, T. (1998, 8-11 Sept.). Correlation model for shadowing in a small urban macro cell. In *9th IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 98)*, pages 1–5, Boston, MA, USA.

[Szyszkowicz et al., 2010] Szyszkowicz, S., Yanikomeroglu, H., and Thompson, J. (2010). On the feasibility of wireless shadowing correlation models. *IEEE Transactions on Vehicular Technology*, 59(9):4222–4236.

[T. Okumura and Fukuda, 1968] T. Okumura, E. O. and Fukuda, K. (1968). Field strength and its variability in VHF and UHF land-mobile radio service. *Rev. of the Electrical Communication Laboratory*, 16:825–873.

[Thomas and Ros, 2005] Thomas, F. and Ros, L. (2005). Revisiting trilateration for robot localization. *IEEE Transactions on Robotics*, 21(1):93–101.

[Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

[Tobler, 1970] Tobler, W. (1970). A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46(2):234–240.

[Torrieri, 1984] Torrieri, D. (1984). Statistical theory of passive location systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):183–198.

[Turin et al., 2002] Turin, W., Jana, R., Ghassemzadeh, S., Rice, C., and Tarokh, T. (2002, 21-23 May). Autoregressive modeling of an indoor UWB channel. In *IEEE Conf. on Ultra Wideband Systems and Technologies*, pages 71–74, Baltimore, USA.

[Uhlenbeck and Ornstein, 1930] Uhlenbeck, G. E. and Ornstein, L. S. (1930). On the theory of the brownian motion. *Phys. Rev.*, 36:823–841.

[Vaghefi and Buehrer, 2013] Vaghefi, R. and Buehrer, R. (2013, 26-31 May). Received signal strength-based sensor localization in spatially correlated shadowing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4076–4080, Vancouver, BC, Canada.

[Vallet et al., 2012] Vallet, J., Kaltiokallio, O., Myrsky, M., Saarinen, J., and Bocca, M. (2012, 27-29 Aug.). Simultaneous RSS-based localization and model calibration in wireless networks with a mobile robot. In *Procedia Computer Science*, volume 10, pages 1106–1113, Niagara Falls, ON, Canada. Elsevier.

International Workshop on Cooperative Robots and Sensor Networks (RoboSense).

[Vallet et al., 2013] Vallet, J., Kaltiokallio, O., Saarinen, J., Myrsky, M., and Bocca, M. (2013, 20-21 Mar.). On the sensitivity of RSS based localization using the log-normal model: an empirical study. In *10th Workshop on Positioning Navigation and Communication (WPNC)*, pages 1–6, Dresden, Germany.

[Vallet García, 2015] Vallet García, J. M. (2015, 22-24 June). MAP RSS based joint localization and model identification under spatially autocorrelated noise. In *IEEE Int. Conf. on Localization and GNSS*, pages 1–8, Gothenburg, Sweden.

[Weitzen and Lowe, 2002] Weitzen, J. and Lowe, T. (2002). Measurement of angular and distance correlation properties of log-normal shadowing at 1900 MHz and its application to design of pcs systems. *IEEE Transactions on Vehicular Technology*, 51(2):265–273.

[Whitteker, 1988] Whitteker, J. (1988). Measurements of path loss at 910 MHz for proposed microcell urban mobile systems. *IEEE Transactions Vehicular Technology*, 37(3):125–129.

[Xia et al., 1992] Xia, H., Bertoni, H. L., Maciel, L., Lindsay-Stewart, A., Rowe, R., and Grindstaff, L. (1992, 10-13 May). Radio propagation measurements and modelling for line-of-sight microcellular systems. In *IEEE 42nd Vehicular Technology Conference*, volume 1, pages 349–354, Denver, CO, USA.

[Yang and Chen, 2009] Yang, J. and Chen, Y. (2009, 30 Nov. - 4 Dec.). Indoor localization using improved RSS-based lateration methods. In *Proc. IEEE Global Telecommunications Conf. GLOBECOM 2009*, pages 1–6, Honolulu, HI, USA.

[Yang and Tsai, 2009] Yang, K.-J. and Tsai, Y.-R. (2009, 5-8 Apr.). Location tracking in mobile networks under correlated shadowing effects. In *IEEE Wireless Communications and Networking Conference*, pages 1–5, Budapest, Hungary.

[Zemek et al., 2008] Zemek, R., Anzai, D., Hara, S., Yanagihara, K., and Kitayama, K.-i. (2008). RSSI-based localization without a prior knowledge of channel model parameters. *International Journal of Wireless Information Networks*, 15:128–136.

[Zemek et al., 2007] Zemek, R., Hara, S., Yanagihara, K., and Kitayama, K.-i. (2007, 3-7 Sept.). A joint estimation of target location and channel model parameters in an IEEE 802.15.4-based wireless sensor network. In *IEEE 18th*

International Symposium on Personal, Indoor and Mobile Radio Communications, pages 1 –5, Athens, Greece.

[Zhu et al., 2011] Zhu, C., Byrd, Richard Nocedal, J., and Morales, J. L. (2011). L-BFGS-B software for large-scale bound-constrained optimization (v3.0). Online: `http://www.ece.northwestern.edu/~nocedal/lbfgsb.html`.

[Žerovnik et al., 2013] Žerovnik, G., Trkov, A., Smith, D. L., and Capote, R. (2013). Transformation of correlation coefficients between normal and lognormal distribution and implications for nuclear applications. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 727(0):33 – 39.

Appendices

Appendix A

Appendices for Chapter 4

A.1 LMMSE Estimator of d^m

Given the RSS measurements in W (denoted by z_{w}), we are looking for an estimator of d^m linear in z_{w}^m that minimizes the MSE. The family of estimators of d^m linear in z_{w}^m can be written as $\widehat{d^m} = \kappa z_{\mathrm{w}}^m$. Thus, the goal is to find the value of κ that minimizes

$$\mathrm{E}\left[(\widehat{d^m} - d^m)^2\right] = \mathrm{E}\left[(\kappa z_{\mathrm{w}}^m - d^m)^2\right]. \tag{A.1}$$

From (4.21), we can write

$$z_{\mathrm{w}} = d\, 10^{-\frac{\varepsilon}{10n}} = d \implies z_{\mathrm{w}}^m = d^m\, 10^{-\frac{m\varepsilon}{10n}} = d^m\, e^{-\frac{m\varepsilon \ln 10}{10n}} = d^m\, e^{-w}, \tag{A.2}$$

where $w \equiv \frac{m\varepsilon \ln 10}{10n}$. Thus, (A.1) becomes

$$\begin{aligned}\mathrm{E}\left[(\kappa z_{\mathrm{w}}^m - d^m)^2\right] &= \mathrm{E}\left[(\kappa d^m e^{-w} - d^m)^2\right] = d^{2m}\mathrm{E}\left[(\kappa e^{-w} - 1)^2\right] \\ &= d^{2m}\mathrm{E}\left[\kappa^2 e^{-2w} - 2\kappa e^{-w} + 1\right] \\ &= d^{2m}\left\{\kappa^2 \mathrm{E}\left[e^{-2w}\right] - 2\kappa \mathrm{E}\left[e^{-w}\right] + 1\right\}. \end{aligned} \tag{A.3}$$

Noting that $w \sim \mathcal{N}(0, \gamma_m)$ (see definitions (4.9) and (4.27)) and that e^{-w} is log-normally distributed, we have

$$\mathrm{E}\left[e^{-w}\right] = e^{\gamma_m^2/2} = \Gamma_m \tag{A.4}$$

$$\mathrm{E}\left[e^{-2w}\right] = e^{(2\gamma_m)^2/2} = e^{4\gamma_m^2/2} = \Gamma_m^4, \tag{A.5}$$

where we have used the definition (4.28). Using this, (A.3) becomes

$$\mathrm{E}\left[(\kappa z_{\mathrm{w}}^m - d^m)^2\right] = d^{2m}\left\{\kappa^2 \Gamma_m^4 - 2\kappa\Gamma_m + 1\right\}. \tag{A.6}$$

Taking derivatives with respect to κ and equating them to zero, we obtain

$$\frac{\partial \mathrm{E}\left[(\kappa z_{\mathrm{w}}^m - d^m)^2\right]}{\partial \kappa} = 2\kappa\Gamma_m^4 - 2\Gamma_m = 0 \implies \kappa = \frac{\Gamma_m}{\Gamma_m^4} = \Gamma_m^{-3} \tag{A.7}$$

and therefore, the LMMSE estimator of d^m is

$$\widehat{d^m} = \frac{z_{\mathrm{w}}^m}{\Gamma_m^3}. \tag{A.8}$$

A.2 Localization using Taylor-Series Linearization Estimation

For the convenience of the reader, let us reproduce the model (4.53) and the cost function (4.54), which are respectively

$$\boldsymbol{z} = \boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) + \boldsymbol{\epsilon} \tag{A.9}$$

$$Q = \boldsymbol{\epsilon}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon} = \left[\boldsymbol{z} - \boldsymbol{f}(\vec{\boldsymbol{s}})\right]^t \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{z} - \boldsymbol{f}(\vec{\boldsymbol{s}})\right]. \tag{A.10}$$

Note that we use $\vec{\boldsymbol{s}}_{\mathrm{t}}$ to denote the true node position, whereas $\vec{\boldsymbol{s}}$ is an independent variable. Equation (A.9) is then the generative process. Our purpose is to find an estimate of $\vec{\boldsymbol{s}}_{\mathrm{t}}$ through finding the $\vec{\boldsymbol{s}}$ that minimizes A.10.

Suppose that we have an initial guess of the position $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$. Expanding $\boldsymbol{f}$ in Taylor's series with respect to $\vec{\boldsymbol{s}}$ about $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$ and keeping only the first degree terms we have

$$\boldsymbol{f}(\vec{\boldsymbol{s}}) \approx \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) + \left.\frac{\partial \boldsymbol{f}(\vec{\boldsymbol{s}})}{\partial \vec{\boldsymbol{s}}}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} (\vec{\boldsymbol{s}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) = \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) + \mathbf{J}_{\mathrm{n}} \vec{\boldsymbol{\delta}}, \tag{A.11}$$

where $\mathbf{J}_{\mathrm{n}}$ denotes the following $K \times 2$ Jacobian,

$$\mathbf{J}_{\mathrm{n}} \equiv \left.\frac{\partial \boldsymbol{f}(\vec{\boldsymbol{s}})}{\partial \vec{\boldsymbol{s}}}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} = \begin{pmatrix} \left.\frac{\partial f_1}{\partial x}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \left.\frac{\partial f_1}{\partial y}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \\ \cdots & \cdots \\ \left.\frac{\partial f_K}{\partial x}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} & \left.\frac{\partial f_K}{\partial y}\right|_{\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}} \end{pmatrix}, \tag{A.12}$$

and $\vec{\boldsymbol{\delta}} = (\vec{\boldsymbol{s}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})$. Substituting A.11 in A.10 we obtain

$$Q = \left[\boldsymbol{z} - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) - \mathbf{J}_{\mathrm{n}} \vec{\boldsymbol{\delta}}\right]^t \boldsymbol{\Sigma}^{-1} \left[\boldsymbol{z} - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) - \mathbf{J}_{\mathrm{n}} \vec{\boldsymbol{\delta}}\right]. \tag{A.13}$$

The model associated with optimizing A.13 is then

$$\boldsymbol{z} - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) = \mathbf{J}_{\mathrm{n}} \vec{\boldsymbol{\delta}},$$

where the optimization is on $\vec{\boldsymbol{\delta}}$ instead of $\vec{\boldsymbol{s}}$ straight. We have then a GLS problem on $\vec{\boldsymbol{\delta}}$. The solution is [Davidson and MacKinnon, 2004, p. 258]

$$\vec{\boldsymbol{\delta}} = (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} [\boldsymbol{z} - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})]. \tag{A.14}$$

Using the definition of $\vec{\boldsymbol{\delta}}$ and solving for $\vec{\boldsymbol{s}}$, we get $\vec{\boldsymbol{s}} = \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} + \vec{\boldsymbol{\delta}}$, which we can use to construct a new position estimate

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}} = \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} + (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} [\boldsymbol{z} - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})]. \tag{A.15}$$

Obviously, we can then continue the iteration and find $\hat{\vec{\boldsymbol{s}}}_{n+2}, \hat{\vec{\boldsymbol{s}}}_{n+3}$, and so forth, until convergence.

In order to study the statistical properties of the estimator, it will be useful to find an alternative expression of A.15 in terms of the true position $\vec{\boldsymbol{s}}_{\mathrm{t}}$. First, substituting $\boldsymbol{z} = \boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) + \boldsymbol{\epsilon}$ for $\boldsymbol{z}$ in A.15 yields

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}} = \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} + (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} [\boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) + \boldsymbol{\epsilon}]. \tag{A.16}$$

Additionally, we can express $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$ as a function of $\vec{\boldsymbol{s}}_{\mathrm{t}}$ as follows:

$$\begin{aligned}
\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} &= \vec{\boldsymbol{s}}_{\mathrm{t}} - \vec{\boldsymbol{s}}_{\mathrm{t}} + \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}} \\
&= \vec{\boldsymbol{s}}_{\mathrm{t}} - (\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) \\
&= \vec{\boldsymbol{s}}_{\mathrm{t}} - \underbrace{(\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}}_{\boldsymbol{I}} (\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}).
\end{aligned}$$

Substituting the previous expression for $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}$ in A.16 leads to

$$\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}} = \vec{\boldsymbol{s}}_t + (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \{[\boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})] - \mathbf{J}_{\mathrm{n}}(\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}}) + \boldsymbol{\epsilon}\}. \tag{A.17}$$

Using this expression it is straightforward to calculate the bias of the estimator:

$$\mathrm{E}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}}\right] - \vec{\boldsymbol{s}}_t = (\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}})^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \{\underbrace{[\boldsymbol{f}(\vec{\boldsymbol{s}}_{\mathrm{t}}) - \boldsymbol{f}(\hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})] - \mathbf{J}_{\mathrm{n}}(\vec{\boldsymbol{s}}_{\mathrm{t}} - \hat{\vec{\boldsymbol{s}}}_{\mathrm{n}})}_{\boldsymbol{e}_{\mathbf{L}}} + \mathrm{E}[\boldsymbol{\epsilon}]\},$$

where $\boldsymbol{e}_{\mathbf{L}}$ is the linearization error. The covariance of $\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}}$ can also be calculated from A.17, resulting in

$$\begin{aligned}
\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}_{\mathrm{n+1}}\right] &= [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathrm{Cov}[\boldsymbol{\epsilon}] \left\{[\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1}\right\}^t \\
&= [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \mathbf{J}_{\mathrm{n}}^t \; \boldsymbol{\Sigma}^{-1} \boldsymbol{\Sigma} \; \left\{[\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1}\right\}^t \\
&= [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \mathbf{J}_{\mathrm{n}}^t \left\{[\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1}\right\}^t \\
&= [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \; \mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}} \; [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1} \\
&= [\mathbf{J}_{\mathrm{n}}^t \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\mathrm{n}}]^{-1}.
\end{aligned} \tag{A.18}$$

A.3 Expectation and Covariance of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ Under the AGN Model

Recall the definition of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ for the AGN model from (4.88):

$$\hat{\boldsymbol{b}}_{\mathrm{L}} = \frac{1}{2}\begin{bmatrix} (\hat{d}_1^2 - \hat{d}_{\mathrm{i}}^2) + (r_{\mathrm{i}}^2 - r_1^2) \\ \cdots \\ (\hat{d}_{\mathrm{i}-1}^2 - \hat{d}_{\mathrm{i}}^2) + (r_{\mathrm{i}}^2 - r_{\mathrm{i}-1}^2) \\ (\hat{d}_{\mathrm{i}+1}^2 - \hat{d}_{\mathrm{i}}^2) + (r_{\mathrm{i}}^2 - r_{\mathrm{i}+1}^2) \\ \cdots \\ (\hat{d}_{\mathrm{K}}^2 - \hat{d}_{\mathrm{i}}^2) + (r_{\mathrm{i}}^2 - r_{\mathrm{K}}^2) \end{bmatrix}. \tag{A.19}$$

The k-th row of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ is

$$\hat{b}_{\mathrm{L_k}} = \frac{1}{2}(\hat{d}_{\mathrm{k}}^2 - \hat{d}_{\mathrm{i}}^2) + \frac{1}{2}(r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2).$$

Recognizing that the second term on the right-hand side is constant, we can write

$$\mathrm{E}\left[\hat{b}_{\mathrm{L_k}}\right] = \frac{1}{2}\mathrm{E}\left[\hat{d}_{\mathrm{k}}^2 - \hat{d}_{\mathrm{i}}^2\right] + \frac{1}{2}(r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2) \tag{A.20}$$

$$\mathrm{Var}\left[\hat{b}_{\mathrm{L_k}}\right] = \frac{1}{4}\mathrm{Var}\left[\hat{d}_{\mathrm{k}}^2 - \hat{d}_{\mathrm{i}}^2\right]. \tag{A.21}$$

From the measurement model, $\hat{d}_{\mathrm{k}} \sim N(d_{\mathrm{k}}, \sigma^2)$, and thus $(\hat{d}_{\mathrm{k}}/\sigma)^2$ is distributed according to the non-central chi-squared distribution with one degree of freedom and non-centrality parameter $\lambda = (d_{\mathrm{k}}/\sigma)^2$. Its mean and covariance are

$$\mathrm{E}\left[(\hat{d}_{\mathrm{k}}/\sigma)^2\right] = 1 + (d_{\mathrm{k}}/\sigma)^2$$
$$\mathrm{Var}\left[(\hat{d}_{\mathrm{k}}/\sigma)^2\right] = 2\left[1 + 2(d_{\mathrm{k}}/\sigma)^2\right],$$

from where we obtain that

$$\mathrm{E}\left[\hat{d}_{\mathrm{k}}^2\right] = d_{\mathrm{k}}^2 + \sigma^2 \tag{A.22}$$

$$\mathrm{Var}\left[\hat{d}_{\mathrm{k}}^2\right] = \sigma^2(4d_{\mathrm{k}}^2 + 2\sigma^2). \tag{A.23}$$

Thus,

$$\begin{aligned}\mathrm{E}\left[\hat{b}_{\mathrm{L_k}}\right] &= \frac{1}{2}\left(d_{\mathrm{k}}^2 + \sigma^2 - (d_{\mathrm{i}}^2 + \sigma^2)\right) + \frac{1}{2}(r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2) \\ &= \frac{1}{2}(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2) + \frac{1}{2}(r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2) = b_{\mathrm{L_k}},\end{aligned}$$

and accordingly,

$$\mathrm{E}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] = \boldsymbol{b}_{\mathbf{L}}. \tag{A.24}$$

To calculate the covariance of the estimator, we depart from A.21, from which we can write

$$\mathrm{Var}\left[\hat{b}_{\mathrm{L_k}}\right] = \frac{1}{4}\mathrm{Var}\left[\hat{d}^2_{\mathrm{k}} - \hat{d}^2_{\mathrm{i}}\right] = \frac{1}{4}\left(\mathrm{Var}\left[\hat{d}^2_{\mathrm{k}}\right] + \mathrm{Var}\left[\hat{d}^2_{\mathrm{i}}\right] - 2\ \mathrm{Cov}\left[\hat{d}^2_{\mathrm{k}}, \hat{d}^2_{\mathrm{i}}\right]\right).$$

Assuming IID measurement errors, $\mathrm{Cov}\left[\hat{d}^2_{\mathrm{k}}, \hat{d}^2_{\mathrm{i}}\right] = 0$. For the vector of measurements $\hat{\boldsymbol{b}}_{\mathrm{L}}$, we then have

$$\mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] = \frac{1}{4}\left(\mathrm{Cov}\left[\begin{pmatrix} \hat{d}^2_1 \\ \dots \\ \hat{d}^2_{\mathrm{i}-1} \\ \hat{d}^2_{\mathrm{i}+1} \\ \dots \\ \hat{d}^2_{\mathrm{K}} \end{pmatrix}\right] + \mathrm{Cov}\left[\mathbf{1}_{\mathbf{L}}\hat{d}^2_i\right]\right), \tag{A.25}$$

where $\mathbf{1}_{\mathbf{L}} = \begin{pmatrix} 1 & \dots & 1 \end{pmatrix}^t$, $\dim(\mathbf{1}_{\mathbf{L}}) = (K-1)\times 1$. Note that $\mathbf{1}_{\mathbf{L}}\hat{d}^2_i$ is a column vector with all the elements equal to $\hat{d}^2_i$. Using A.23, then we can write

$$\mathrm{Cov}\left[\begin{pmatrix} \hat{d}^2_1 \\ \dots \\ \hat{d}^2_{\mathrm{i}-1} \\ \hat{d}^2_{\mathrm{i}+1} \\ \dots \\ \hat{d}^2_{\mathrm{K}} \end{pmatrix}\right] = \sigma^2(4\,\boldsymbol{D}^2_{\mathbf{L}} + 2\sigma^2\mathbf{I})$$

$$\mathrm{Cov}\left[\mathbf{1}_{\mathbf{L}}\hat{d}^2_i\right] = \mathbf{1}_{\mathbf{L}}\ \mathrm{Cov}\left[\hat{d}^2_i\right]\ \mathbf{1}^t_{\mathbf{L}} = \sigma^2(4d^2_{\mathrm{i}} + 2\sigma^2)\ \mathbf{1}_{\mathbf{L}}\mathbf{1}^t_{\mathbf{L}},$$

where $\boldsymbol{D}_{\mathbf{L}} = \mathrm{diag}(d_1, \dots, d_{\mathrm{i}-1}, d_{\mathrm{i}+1}, \dots, d_{\mathrm{K}})$. Finally, the covariance of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ becomes

$$\boldsymbol{\Sigma}_{\mathbf{L}} \equiv \mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] = \sigma^2(\boldsymbol{D}^2_{\mathbf{L}} + d^2_{\mathrm{i}}\mathbf{1}_{\mathbf{L}}\mathbf{1}^t_{\mathbf{L}}) + \frac{\sigma^2}{2}(\mathbf{I} + \mathbf{1}_{\mathbf{L}}\mathbf{1}^t_{\mathbf{L}}). \tag{A.26}$$

A.4 Expectation and Covariance of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ Under the Log-Normal Model

Under the log-normal model we use square distance estimates instead of distance estimates in the matrix $\boldsymbol{b}_{\mathrm{L}}$, leading to

$$\hat{\boldsymbol{b}}_{\mathrm{L}} = \frac{1}{2}\begin{bmatrix} (\widehat{d_1^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_1^2) \\ \cdots \\ (\widehat{d_{\mathrm{i-1}}^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_{\mathrm{i-1}}^2) \\ (\widehat{d_{\mathrm{i+1}}^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{i}}^2 - r_{\mathrm{i+1}}^2) \\ \cdots \\ (\widehat{d_{\mathrm{K}}^2} - \widehat{d_{\mathrm{i}}^2}) + (r_{\mathrm{K}}^2 - r_{\mathrm{K}}^2) \end{bmatrix}. \tag{A.27}$$

We calculate now the expectation and covariance of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ when using the different distance estimators.

ML RSS Distance Estimator

From (4.11), the ML distance estimator can be expressed as

$$\hat{d}_{\mathrm{ML}} = d\, e^{\epsilon'}, \;\; \epsilon' \sim N(0, \gamma^2)$$

which is log-normally distributed. The square of this estimator is then

$$\hat{d}_{\mathrm{ML}}^2 = d^2\, e^{2\epsilon'} = d^2\, e^{\epsilon''}, \;\; \epsilon'' \sim N(0, (2\gamma)^2),$$

which is also log-normally distributed. Thus,

$$\mathrm{E}\left[\hat{d}_{\mathrm{ML}}^2\right] = d^2 e^{(2\gamma)^2/2} = d^2 e^{2\gamma^2} = d^2\Gamma^4 = d^2\Gamma_2 \tag{A.28}$$

$$\mathrm{Var}\left[\hat{d}_{\mathrm{ML}}^2\right] = d^4(e^{(2\gamma)^2} - 1)e^{(2\gamma)^2} = d^4\Gamma^8(\Gamma^8 - 1) = d^4\Gamma_2^2(\Gamma_2^2 - 1), \tag{A.29}$$

where Γ and Γ_m are defined as in (4.14) and (4.28), respectively. These are effectively the same expressions as (4.29) and (4.30) with $m = 2$. The expectation of the k-th row of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ then becomes

$$\begin{aligned} \mathrm{E}\left[\hat{b}_{\mathrm{L_k}}\right] &= \frac{1}{2}\left[(d_{\mathrm{k}}^2\Gamma_2 - d_{\mathrm{i}}^2\Gamma_2) + (r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2)\right] \\ &= \underbrace{\frac{1}{2}\left[(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2)\Gamma_2 - (d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2)\right]}_{\frac{1}{2}(\Gamma_2 - 1)(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2)} + \underbrace{\frac{1}{2}\left[(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2) + (r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2)\right]}_{b_{\mathrm{L_k}}} \\ &= \frac{1}{2}(\Gamma_2 - 1)(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2) + b_{\mathrm{L_k}}, \end{aligned}$$

and the expectation of $\hat{\boldsymbol{b}}_{\text{L}}$

$$\text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{2}(\Gamma_2 - 1)\begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix} + \boldsymbol{b}_{\text{L}}.$$

From A.27, the covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ can be expressed as:

$$\text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{4}\left(\text{Cov}\left[\begin{pmatrix} \widehat{d_1^2} \\ \cdots \\ \widehat{d_{\text{i}-1}^2} \\ \widehat{d_{\text{i}+1}^2} \\ \cdots \\ \widehat{d_{\text{K}}^2} \end{pmatrix}\right] + \text{Cov}\left[\mathbf{1}_{\text{L}}\widehat{d_{\text{i}}^2}\right]\right), \tag{A.30}$$

where $\mathbf{1}_{\text{L}} = \begin{pmatrix}1 & \dots & 1\end{pmatrix}^t$, $\dim(\mathbf{1}_{\text{L}}) = (K-1)\times 1$. Using A.29 and assuming IID measurements, we can write

$$\text{Cov}\left[\begin{pmatrix} \widehat{d_1^2} \\ \cdots \\ \widehat{d_{\text{i}-1}^2} \\ \widehat{d_{\text{i}+1}^2} \\ \cdots \\ \widehat{d_{\text{K}}^2} \end{pmatrix}\right] = \Gamma_2^2(\Gamma_2^2 - 1)\boldsymbol{D}_{\text{L}}^4$$

$$\text{Cov}\left[\mathbf{1}_{\text{L}}\hat{d}_i^2\right] = \mathbf{1}_{\text{L}}\,\text{Cov}\left[\hat{d}_i^2\right]\,\mathbf{1}_{\text{L}}^t = \Gamma_2^2(\Gamma_2^2 - 1)d_{\text{i}}^4\,\mathbf{1}_{\text{L}}\mathbf{1}_{\text{L}}^t,$$

where $\boldsymbol{D}_{\text{L}} = \text{diag}(d_1 \dots d_{\text{i}-1}, d_{\text{i}+1}, \dots, d_{\text{K}})$. Finally, the covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ becomes

$$\boldsymbol{\Sigma}_{\text{L}} \equiv \text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{4}\Gamma_2^2(\Gamma_2^2 - 1)\left(\boldsymbol{D}_{\text{L}}^4 + d_{\text{i}}^4\,\mathbf{1}_{\text{L}}\mathbf{1}_{\text{L}}^t\right).$$

Recall that the OLS position estimator is

$$\hat{\vec{\boldsymbol{s}}} = \boldsymbol{C}_{\text{L}}\hat{\boldsymbol{b}}_{\text{L}}, \text{ where } \boldsymbol{C}_{\text{L}} = [\boldsymbol{A}_{\text{L}}^t\boldsymbol{A}_{\text{L}}]^{-1}\boldsymbol{A}_{\text{L}}^t$$

The expectation of the OLS position estimator then becomes

$$\begin{aligned}\text{E}\left[\hat{\vec{\boldsymbol{s}}}\right] &= \boldsymbol{C}_{\text{L}}\text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{2}(\Gamma_2 - 1)\boldsymbol{C}_{\text{L}} \begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix} + \underbrace{\boldsymbol{C}_{\text{L}}\boldsymbol{b}_{\text{L}}}_{\vec{\boldsymbol{s}}} \\ &= \vec{\boldsymbol{s}} + \frac{1}{2}(\Gamma_2 - 1)\boldsymbol{C}_{\text{L}} \begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix}\end{aligned}$$

which, as we can see, is biased. The covariance of the OLS position estimator is

$$\text{Cov}\left[\hat{\vec{\boldsymbol{s}}}\right] = \boldsymbol{C}_{\text{L}}\boldsymbol{\Sigma}_{\text{L}}\boldsymbol{C}_{\text{L}}^t.$$

Unbiased RSS Squared Distance Estimator

From (4.32), the unbiased squared distance estimator can be expressed as

$$\hat{P}_{\text{U}} = \frac{\hat{d}_{\text{ML}}^2}{\Gamma_2}. \tag{A.31}$$

From (4.33), its variance is:

$$\text{Var}\left[\hat{P}_{\text{U}}\right] = d^4(\Gamma_2^2 - 1), \tag{A.32}$$

where Γ_m is defined as in (4.28). The expectation of the k-th row of $\hat{\boldsymbol{b}}_{\text{L}}$ (see A.27) becomes

$$\text{E}\left[\hat{b}_{\text{L}_{\text{k}}}\right] = \frac{1}{2}\left[(d_{\text{k}}^2 - d_{\text{K}}^2) + (r_{\text{K}}^2 - r_{\text{k}}^2)\right] = b_{\text{L}_{\text{k}}},$$

and therefore $\text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \boldsymbol{b}_{\text{L}}$ and the position estimator is unbiased.

The covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ can be calculated using A.30. Using A.32 and assuming

IID measurements, we can write

$$\mathrm{Cov}\left[\begin{pmatrix} \widehat{d_1^2} \\ \cdots \\ \widehat{d_{i-1}^2} \\ \widehat{d_{i+1}^2} \\ \cdots \\ \widehat{d_K^2} \end{pmatrix}\right] = (\Gamma_2^2 - 1)\boldsymbol{D}_{\mathbf{L}}^4$$

$$\mathrm{Cov}\left[\mathbf{1}_{\mathbf{L}}\hat{d}_i^2\right] = \mathbf{1}_{\mathbf{L}}\,\mathrm{Cov}\left[\hat{d}_i^2\right]\,\mathbf{1}_{\mathbf{L}}^t = (\Gamma_2^2 - 1)d_i^4\,\mathbf{1}_{\mathbf{L}}\mathbf{1}_{\mathbf{L}}^t$$

Finally, the covariance of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ becomes

$$\boldsymbol{\Sigma}_{\mathbf{L}} \equiv \mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right] = \frac{1}{4}(\Gamma_2^2 - 1)\big(\boldsymbol{D}_{\mathbf{L}}^4 + d_i^4\,\mathbf{1}_{\mathbf{L}}\mathbf{1}_{\mathbf{L}}^t\big),$$

and the covariance of the OLS position estimator is

$$\mathrm{Cov}\left[\hat{\vec{\boldsymbol{s}}}\right] = \boldsymbol{C}_{\mathbf{L}}\boldsymbol{\Sigma}_{\mathbf{L}}\boldsymbol{C}_{\mathbf{L}}^t.$$

LMMS RSS Squared Distance Estimator

From (4.35), the LMMSE squared distance estimator can be expressed as

$$\hat{P}_{\mathrm{V}} = \frac{\hat{d}_{\mathrm{ML}}^2}{\Gamma_2^3}. \tag{A.33}$$

From (4.36) and (4.37), its expectation and variance are:

$$\mathrm{E}\left[\hat{P}_{\mathrm{V}}\right] = d^2\Gamma_2^{-2} \tag{A.34}$$

$$\mathrm{Var}\left[\hat{P}_{\mathrm{V}}\right] = d^4\frac{\Gamma_2^2 - 1}{\Gamma_2^4}. \tag{A.35}$$

The expectation of the k-th row of $\hat{\boldsymbol{b}}_{\mathrm{L}}$ (see A.27) becomes

$$\begin{aligned}
\mathrm{E}\left[\hat{b}_{\mathrm{L_k}}\right] &= \frac{1}{2}\left[(d_{\mathrm{k}}^2\Gamma_2^{-2} - d_{\mathrm{i}}^2\Gamma_2^{-2}) + (r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2)\right] \\
&= \underbrace{\frac{1}{2}\left[(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2)\Gamma_2^{-2} - (d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2)\right]}_{\frac{1}{2}(\Gamma_2^{-2}-1)(d_{\mathrm{k}}^2-d_{\mathrm{i}}^2)} + \underbrace{\frac{1}{2}\left[(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2) + (r_{\mathrm{i}}^2 - r_{\mathrm{k}}^2)\right]}_{b_{\mathrm{L_k}}} \\
&= \frac{1}{2}(\Gamma_2^{-2} - 1)(d_{\mathrm{k}}^2 - d_{\mathrm{i}}^2) + b_{\mathrm{L_k}},
\end{aligned}$$

and the expectation of $\hat{\boldsymbol{b}}_{\text{L}}$ becomes

$$\text{E}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{2}(\Gamma_2^{-2} - 1) \begin{pmatrix} d_1^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{i}-1}^2 - d_{\text{i}}^2 \\ d_{\text{i}+1}^2 - d_{\text{i}}^2 \\ \cdots \\ d_{\text{K}}^2 - d_{\text{i}}^2 \end{pmatrix} + \boldsymbol{b}_{\text{L}}.$$

The covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ can be calculated using A.30. Assuming IID measurements, we can write

$$\text{Cov}\left[\begin{pmatrix} \widehat{d_1^2} \\ \cdots \\ \widehat{d_{\text{i}-1}^2} \\ \widehat{d_{\text{i}+1}^2} \\ \cdots \\ \widehat{d_{\text{K}}^2} \end{pmatrix}\right] = \frac{\Gamma_2^2 - 1}{\Gamma_2^4} \boldsymbol{D}_{\text{L}}^4$$

$$\text{Cov}\left[\mathbf{1}_{\text{L}} \hat{d}_i^2\right] = \mathbf{1}_{\text{L}} \, \text{Cov}\left[\hat{d}_i^2\right] \, \mathbf{1}_{\text{L}}^t = \frac{\Gamma_2^2 - 1}{\Gamma_2^4} d_{\text{i}}^4 \, \mathbf{1}_{\text{L}} \mathbf{1}_{\text{L}}^t.$$

Finally, the covariance of $\hat{\boldsymbol{b}}_{\text{L}}$ becomes

$$\boldsymbol{\Sigma}_{\text{L}} \equiv \text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{4} \frac{\Gamma_2^2 - 1}{\Gamma_2^4} \left(\boldsymbol{D}_{\text{L}}^4 + d_{\text{i}}^4 \, \mathbf{1}_{\text{L}} \mathbf{1}_{\text{L}}^t\right),$$

and the covariance of the OLS position estimator is:

$$\text{Cov}\left[\hat{\vec{\boldsymbol{s}}}\right] = \boldsymbol{C}_{\text{L}} \boldsymbol{\Sigma}_{\text{L}} \boldsymbol{C}_{\text{L}}^t.$$

Appendix B

Appendices for Chapter 6

B.1 Derivation of FIM for Figure 6.3

In view of Figure 6.3, from (6.24) and with $\vec{\boldsymbol{s}} = (0,0)^t$, we get

$$\mathbf{FIM} = \left(\frac{10n}{\ln 10}\right)^2 \frac{1}{d^4}\, \boldsymbol{R}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{R}.$$

Now, the covariance matrix (6.23) can be expressed as:

$$\boldsymbol{\Sigma} = \sigma^2 \left[\alpha \mathbf{I} + (1-\alpha)\boldsymbol{W}\right] = \sigma^2 \begin{pmatrix} 1 & a \\ a & 1 \end{pmatrix},\ a = (1-\alpha)e^{-l/l_0},$$

where $l = 2d$ in our case. The inverse of $\boldsymbol{\Sigma}$ is

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma^2(1-a^2)} \begin{pmatrix} 1 & -a \\ -a & 1 \end{pmatrix}.$$

Using $\vec{\boldsymbol{r}}_1 = (d,0)^t$ and $\vec{\boldsymbol{r}}_2 = (-d,0)^t$ we have

$$\boldsymbol{R} = \begin{pmatrix} \vec{\boldsymbol{r}}_1 & \vec{\boldsymbol{r}}_2 \end{pmatrix}^t = \begin{pmatrix} d & 0 \\ -d & 0 \end{pmatrix},$$

and thus,

$$\boldsymbol{R}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{R} = \frac{1}{\sigma^2(1-a^2)} \begin{pmatrix} 2d^2(1+a) & 0 \\ 0 & 0 \end{pmatrix} = \frac{2d^2}{\sigma^2(1-a)} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Finally,

$$\mathbf{FIM} = \left(\frac{10n}{\ln 10}\right)^2 \frac{1}{d^4}\, \frac{2d^2}{\sigma^2(1-a)} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{2}{d^2(1-a)} \left(\frac{10n}{\sigma \ln 10}\right)^2 \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

B.2 Derivation of FIM for Figure 6.4a

In view of Figure 6.4a, from (6.24), with $\vec{\boldsymbol{s}} = (0,0)^t$ we get

$$\mathbf{FIM} = \left(\frac{10n}{\ln 10}\right)^2 \frac{1}{d^4}\, \boldsymbol{R}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{R}. \tag{B.1}$$

Now, the covariance matrix (6.23) can be expressed as:

$$\boldsymbol{\Sigma} = \sigma^2 \left[\alpha \mathbf{I} + (1-\alpha)\boldsymbol{W}\right] = \sigma^2 \begin{pmatrix} 1 & a & a \\ a & 1 & b \\ a & b & 1 \end{pmatrix},$$

where

$$a = (1-\alpha)e^{-l_{12}/l_0},\ l_{12} = 2d|cos(\phi/2)|$$
$$b = (1-\alpha)e^{-l_{23}/l_0},\ l_{23} = 2d|sin(\phi)|.$$

Its inverse is

$$\boldsymbol{\Sigma}^{-1} = \frac{1}{\sigma^2(b-1)(2a^2-1-b)} \begin{pmatrix} 1-b^2 & ab-a & ab-a \\ ab-a & 1-a^2 & a^2-b \\ ab-a & a^2-b & 1-a^2 \end{pmatrix}.$$

Using $\vec{\boldsymbol{r}}_1 = (-d,0)^t$, $\vec{\boldsymbol{r}}_2 = d(\cos(\phi), \sin(\phi))^t$ and $\vec{\boldsymbol{r}}_3 = d(\cos(\phi), -\sin(\phi))^t$, we have

$$\boldsymbol{R} = \begin{pmatrix} \vec{\boldsymbol{r}}_1 & \vec{\boldsymbol{r}}_2 \end{pmatrix}^t = d \begin{pmatrix} -1 & 0 \\ \cos(\phi) & \sin(\phi) \\ \cos(\phi) & -\sin(\phi) \end{pmatrix} = \begin{pmatrix} \boldsymbol{x} & \boldsymbol{y} \end{pmatrix},$$

where

$$\boldsymbol{x} = d \begin{pmatrix} -1 \\ \cos(\phi) \\ \cos(\phi) \end{pmatrix},\ \boldsymbol{y} = d \begin{pmatrix} 0 \\ \sin(\phi) \\ -\sin(\phi) \end{pmatrix}.$$

Using this, we can express

$$\boldsymbol{R}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{R} = \begin{pmatrix} \boldsymbol{x}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{x} & \boldsymbol{x}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{y} \\ \boldsymbol{y}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{x} & \boldsymbol{y}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{y} \end{pmatrix}.$$

After some calculations, it can be shown that

$$\begin{aligned} \boldsymbol{x}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{x} &= \frac{d^2}{\sigma^2} \frac{1 + b + 4a\cos(\phi) + 2\cos^2(\phi)}{b + 1 - 2a^2} \\ \boldsymbol{y}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{y} &= \frac{2d^2 \sin^2(\phi)}{\sigma^2(1-b)} \\ \boldsymbol{x}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{y} &= 0. \end{aligned} \tag{B.2}$$

Finally, combining (B.1) and (B.2) we have

$$\mathbf{FIM} = \begin{pmatrix} F_{xx} & 0 \\ 0 & F_{yy} \end{pmatrix}$$

where

$$F_{xx} = \left(\frac{10n}{d\sigma \ln 10}\right)^2 \frac{1 + b + 4a\cos(\phi) + 2\cos^2(\phi)}{b - 2a^2 + 1}$$

$$F_{yy} = \left(\frac{10n}{d\sigma \ln 10}\right)^2 \frac{2\sin^2(\phi)}{1 - b}$$

$$a = (1-\alpha)\exp\left\{-\frac{2d\,|\cos(\phi/2)|}{l_0}\right\}$$

$$b = (1-\alpha)\exp\left\{-\frac{2d\,|\sin(\phi)|}{l_0}\right\}.$$

B.3 Derivation of Cov $[\boldsymbol{b}_{\text{L}}]$ for LLS Estimators

In this appendix we give a detailed derivation of the formulas of Cov $\left[\hat{\boldsymbol{b}}_{\text{L}}\right]$ needed for the LLS position estimators using the log-normal model with spatially correlated noise and autocovariance function defined in (6.15).

Let us suppose that we choose the i-th distance measurement as pivot. From (4.92) we can rewrite $\hat{\boldsymbol{b}}_{\text{L}}$ as:

$$\hat{\boldsymbol{b}}_{\text{L}} = \frac{1}{2}\left(\hat{\boldsymbol{G}} + \boldsymbol{T}\right), \text{ where } \hat{\boldsymbol{G}} = \begin{pmatrix} \widehat{d_1^2} - \widehat{d_{\text{i}}^2} \\ \cdots \\ \widehat{d_{\text{i}-1}^2} - \widehat{d_{\text{i}}^2} \\ \widehat{d_{\text{i}+1}^2} - \widehat{d_{\text{i}}^2} \\ \cdots \\ \widehat{d_{\text{K}}^2} - \widehat{d_{\text{i}}^2} \end{pmatrix} \text{ and } \boldsymbol{T} = \begin{pmatrix} r_{\text{i}}^2 - r_1^2 \\ \cdots \\ r_{\text{i}}^2 - r_{\text{i}-1}^2 \\ r_{\text{i}}^2 - r_{\text{i}+1}^2 \\ \cdots \\ r_{\text{i}}^2 - r_{\text{K}}^2 \end{pmatrix}, \tag{B.3}$$

according to which

$$\text{Cov}\left[\hat{\boldsymbol{b}}_{\text{L}}\right] = \frac{1}{4}\text{Cov}\left[\hat{\boldsymbol{G}}\right]. \tag{B.4}$$

Thus, we need to calculate Cov $\left[\hat{\boldsymbol{G}}\right]$, which has dimension $(K-1)\times(K-1)$. This can be calculated from the covariance of the complete vector of estimates $(\widehat{d_1^2}, \ldots, \widehat{d_{\text{K}}^2})^t$, which we will calculate in the following.

Following the notational scheme of 4.2.3, let us call $\hat{P}_{\text{i}}$ the estimate of d_{i}^m. The complete vector of estimates for the m-th power of the K distances will

then be $\hat{\boldsymbol{P}} = (\hat{P}_1, \ldots, \hat{P}_{\mathrm{K}})^t$. Let us further call $\breve{\boldsymbol{P}}_{\boldsymbol{u}} = (\hat{P}_1, \ldots, \hat{P}_{\mathrm{i}-1})^t$ and $\breve{\boldsymbol{P}}_{\boldsymbol{l}} = (\hat{P}_{\mathrm{i}+1}, \ldots, \hat{P}_{\mathrm{K}})^t$, so that $\hat{\boldsymbol{P}} = (\breve{\boldsymbol{P}}_{\boldsymbol{u}}^t, \hat{P}_{\mathrm{i}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}^t)^t$ is the complete distance powers estimates vector. That is:

$$\breve{\boldsymbol{P}}_{\boldsymbol{u}} \equiv \begin{pmatrix} \hat{P}_1 \\ \vdots \\ \hat{P}_{\mathrm{i}-1} \end{pmatrix} \quad \text{and} \quad \breve{\boldsymbol{P}}_{\boldsymbol{l}} \equiv \begin{pmatrix} \hat{P}_{\mathrm{i}+1} \\ \vdots \\ \hat{P}_{\mathrm{K}} \end{pmatrix}, \quad \text{leading to} \quad \hat{\boldsymbol{P}} = \begin{pmatrix} \breve{\boldsymbol{P}}_{\boldsymbol{u}} \\ \cdots\cdots \\ \hat{P}_{\mathrm{i}} \\ \cdots\cdots \\ \breve{\boldsymbol{P}}_{\boldsymbol{l}} \end{pmatrix}. \tag{B.5}$$

Additionally, we call

$$\breve{\boldsymbol{P}} \equiv \begin{pmatrix} \breve{\boldsymbol{P}}_{\boldsymbol{u}} \\ \cdots\cdots \\ \breve{\boldsymbol{P}}_{\boldsymbol{l}} \end{pmatrix}.$$

Using this, we can write[1] $\hat{\boldsymbol{G}} = \breve{\boldsymbol{P}} - \hat{P}_{\mathrm{i}}\mathbf{1}$.

The covariance of $\hat{\boldsymbol{G}}$ can be calculated as

$$\begin{aligned} \mathrm{Cov}\left[\hat{\boldsymbol{G}}\right] &= \mathrm{E}\left[(\hat{\boldsymbol{G}} - \mathrm{E}\left[\hat{\boldsymbol{G}}\right])(\hat{\boldsymbol{G}} - \mathrm{E}\left[\hat{\boldsymbol{G}}\right])^t\right] \\ &= \mathrm{E}\left[\hat{\boldsymbol{G}}\hat{\boldsymbol{G}}^t\right] - \mathrm{E}\left[\hat{\boldsymbol{G}}\right]\mathrm{E}\left[\hat{\boldsymbol{G}}\right]^t. \end{aligned}$$

On the one hand,

$$\begin{aligned} \hat{\boldsymbol{G}}\hat{\boldsymbol{G}}^t &= (\breve{\boldsymbol{P}} - \hat{P}_{\mathrm{i}}\mathbf{1})(\breve{\boldsymbol{P}} - \hat{P}_{\mathrm{i}}\mathbf{1})^t \\ &= \breve{\boldsymbol{P}}(\breve{\boldsymbol{P}} - \hat{P}_{\mathrm{i}}\mathbf{1})^t - \hat{P}_{\mathrm{i}}\mathbf{1}(\breve{\boldsymbol{P}} - \hat{P}_{\mathrm{i}}\mathbf{1})^t \\ &= \breve{\boldsymbol{P}}\breve{\boldsymbol{P}}^t - \underbrace{\breve{\boldsymbol{P}}\mathbf{1}^t\hat{P}_{\mathrm{i}}}_{\boldsymbol{H}} - \underbrace{\hat{P}_{\mathrm{i}}\mathbf{1}\breve{\boldsymbol{P}}^t}_{\boldsymbol{H}^t} + \hat{P}_{\mathrm{i}}\mathbf{1}\mathbf{1}^t\hat{P}_{\mathrm{i}} \\ &= \breve{\boldsymbol{P}}\breve{\boldsymbol{P}}^t - (\boldsymbol{H} + \boldsymbol{H}^t) + \hat{P}_{\mathrm{i}}\mathbf{1}\mathbf{1}^t\hat{P}_{\mathrm{i}}, \end{aligned}$$

which leads to

$$\mathrm{E}\left[\hat{\boldsymbol{G}}\hat{\boldsymbol{G}}^t\right] = \mathrm{E}\left[\breve{\boldsymbol{P}}\breve{\boldsymbol{P}}^t\right] - \mathrm{E}\left[\boldsymbol{H} + \boldsymbol{H}^t\right] + \mathrm{E}\left[\hat{P}_{\mathrm{i}}\mathbf{1}\mathbf{1}^t\hat{P}_{\mathrm{i}}\right].$$

[1] For notational simplicity, in this demonstration we will use the symbol $\mathbf{1}$ to denote the row vector with all its elements equal to one and of the appropriate dimension, which can be derived from the context.

Additionally, $\mathrm{E}\left[\hat{\boldsymbol{G}}\right] = \mathrm{E}\left[\breve{\boldsymbol{P}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}$, and therefore,

$$\begin{aligned}
\mathrm{E}\left[\hat{\boldsymbol{G}}\right]\mathrm{E}\left[\hat{\boldsymbol{G}}\right]^t &= \left(\mathrm{E}\left[\breve{\boldsymbol{P}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\right)\left(\mathrm{E}\left[\breve{\boldsymbol{P}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\right)^t \\
&= \mathrm{E}\left[\breve{\boldsymbol{P}}\right]\left(\mathrm{E}\left[\breve{\boldsymbol{P}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\right)^t - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1})\left(\mathrm{E}\left[\breve{\boldsymbol{P}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\right)^t \\
&= \mathrm{E}\left[\breve{\boldsymbol{P}}\right]\mathrm{E}\left[\breve{\boldsymbol{P}}\right]^t - \underbrace{\mathrm{E}\left[\breve{\boldsymbol{P}}\right]\mathbf{1}^t\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]}_{\bar{\boldsymbol{H}}} - \underbrace{\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\mathrm{E}\left[\breve{\boldsymbol{P}}\right]^t}_{\bar{\boldsymbol{H}}^t} + \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\mathbf{1}^t\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right] \\
&= \mathrm{E}\left[\breve{\boldsymbol{P}}\right]\mathrm{E}\left[\breve{\boldsymbol{P}}\right]^t - (\bar{\boldsymbol{H}} + \bar{\boldsymbol{H}}^t) + \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\mathbf{1}^t\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right].
\end{aligned}$$

With all this we have

$$\begin{aligned}
\mathrm{E}\left[\hat{\boldsymbol{G}}\hat{\boldsymbol{G}}^t\right] - \mathrm{E}\left[\hat{\boldsymbol{G}}\right]\mathrm{E}\left[\hat{\boldsymbol{G}}\right]^t &= \mathrm{E}\left[\breve{\boldsymbol{P}}\breve{\boldsymbol{P}}^t\right] - \mathrm{E}\left[\boldsymbol{H} + \boldsymbol{H}^t\right] + \mathrm{E}\left[\hat{P}_{\mathrm{i}}\mathbf{1}\mathbf{1}^t\hat{P}_{\mathrm{i}}\right] \\
&\quad - \left\{\mathrm{E}\left[\breve{\boldsymbol{P}}\right]\mathrm{E}\left[\breve{\boldsymbol{P}}\right]^t - (\bar{\boldsymbol{H}} + \bar{\boldsymbol{H}}^t) + \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\mathbf{1}^t\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\right\} \\
= \underbrace{\mathrm{E}\left[\breve{\boldsymbol{P}}\breve{\boldsymbol{P}}^t\right] - \mathrm{E}\left[\breve{\boldsymbol{P}}\right]\mathrm{E}\left[\breve{\boldsymbol{P}}\right]^t}_{\mathrm{Cov}\left[\breve{\boldsymbol{P}}\right]} &- \left(\mathrm{E}\left[\boldsymbol{H} + \boldsymbol{H}^t\right] - \left(\bar{\boldsymbol{H}} + \bar{\boldsymbol{H}}^t\right)\right) + \mathbf{1}\mathbf{1}^t\underbrace{\left(\mathrm{E}\left[\hat{P}_{\mathrm{i}}^2\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]^2\right)}_{\mathrm{Cov}\left[\hat{P}_{\mathrm{i}}\right]} \\
= \mathrm{Cov}\left[\breve{\boldsymbol{P}}\right] &- \left(\mathrm{E}\left[\boldsymbol{H} + \boldsymbol{H}^t\right] - \left(\bar{\boldsymbol{H}} + \bar{\boldsymbol{H}}^t\right)\right) + \mathbf{1}\mathbf{1}^t\mathrm{Cov}\left[\hat{P}_{\mathrm{K}}\right].
\end{aligned}$$

Additionally,

$$\mathrm{E}\left[\boldsymbol{H} + \boldsymbol{H}^t\right] - \left(\bar{\boldsymbol{H}} + \bar{\boldsymbol{H}}^t\right) = \mathrm{E}\left[\boldsymbol{H}\right] - \bar{\boldsymbol{H}} + \mathrm{E}\left[\boldsymbol{H}^t\right] - \bar{\boldsymbol{H}}^t, \tag{B.6}$$

and

$$\begin{aligned}
\mathrm{E}\left[\boldsymbol{H}\right] - \bar{\boldsymbol{H}} &= \mathrm{E}\left[\breve{\boldsymbol{P}}\mathbf{1}^t\hat{P}_{\mathrm{i}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}^t\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right] = \left(\mathrm{E}\left[\breve{\boldsymbol{P}}\hat{P}_{\mathrm{i}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\right)\mathbf{1}^t \\
\mathrm{E}\left[\boldsymbol{H}^t\right] - \bar{\boldsymbol{H}}^t &= \mathrm{E}\left[\hat{P}_{\mathrm{i}}\mathbf{1}\breve{\boldsymbol{P}}^t\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathbf{1}\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right] = \mathbf{1}\left(\mathrm{E}\left[\hat{P}_{\mathrm{i}}\breve{\boldsymbol{P}}\right] - \mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\mathrm{E}\left[\hat{P}_{\mathrm{i}}\right]\right).
\end{aligned} \tag{B.7}$$

Note now that the covariance of $\hat{\boldsymbol{P}}$ can be written in blocks as follows:

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}\right] = \left(\begin{array}{c:c:c} \beta_{uu} & \beta_{ui} & \beta_{ul} \\ \hdashline \beta_{iu} & \beta_{ii} & \beta_{il} \\ \hdashline \beta_{lu} & \beta_{li} & \beta_{ll} \end{array}\right) = \left(\begin{array}{c:c:c} \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] & \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \hat{P}_{\mathrm{i}}\right] & \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}\right] \\ \hdashline \mathrm{Cov}\left[\hat{P}_{\mathrm{i}}, \breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] & \mathrm{Cov}\left[\hat{P}_{\mathrm{i}}, \hat{P}_{\mathrm{i}}\right] & \mathrm{Cov}\left[\hat{P}_{\mathrm{i}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}\right] \\ \hdashline \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{l}}, \breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] & \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{l}}, \hat{P}_{\mathrm{i}}\right] & \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{l}}, \breve{\boldsymbol{P}}_{\boldsymbol{l}}\right] \end{array}\right). \tag{B.8}$$

From this matrix we can identify:

$$\begin{aligned}
\mathrm{Cov}\left[\breve{\boldsymbol{P}},\breve{\boldsymbol{P}}\right] &= \left(\begin{array}{c:c} \beta_{uu} & \beta_{ul} \\ \hdashline \beta_{lu} & \beta_{ll} \end{array}\right) \\
\mathrm{Cov}\left[\hat{P}_{\mathrm{i}},\hat{P}_{\mathrm{i}}\right] &= \beta_{ii} \\
\mathrm{Cov}\left[\breve{\boldsymbol{P}},\hat{P}_{\mathrm{i}}\right] &= \left(\begin{array}{c} \beta_{ui} \\ \hdashline \beta_{li} \end{array}\right) \\
\mathrm{Cov}\left[\hat{P}_{\mathrm{i}},\breve{\boldsymbol{P}}\right] &= \left(\begin{array}{c:c} \beta_{iu} & \beta_{il} \end{array}\right).
\end{aligned}$$

Using this, we can express (B.7) as:

$$\begin{aligned}
\mathrm{E}\left[\boldsymbol{H}\right] - \bar{\boldsymbol{H}} &= \left(\begin{array}{c} \beta_{ui} \\ \hdashline \beta_{li} \end{array}\right)\mathbf{1}^t \\
\mathrm{E}\left[\boldsymbol{H}^t\right] - \bar{\boldsymbol{H}}^t &= \mathbf{1}\left(\begin{array}{c:c} \beta_{iu} & \beta_{il} \end{array}\right),
\end{aligned}$$

with which (B.6) becomes

$$\mathrm{E}\left[\boldsymbol{H}+\boldsymbol{H}^t\right] - \left(\bar{\boldsymbol{H}}+\bar{\boldsymbol{H}}^t\right) = \left(\begin{array}{c} \beta_{ui} \\ \hdashline \beta_{li} \end{array}\right)\mathbf{1}^t + \mathbf{1}\left(\begin{array}{c:c} \beta_{iu} & \beta_{il} \end{array}\right). \tag{B.9}$$

Finally, using (B.8) and (B.9) the covariance of $\hat{\boldsymbol{G}}$ can be expressed as

$$\begin{aligned}
\mathrm{Cov}\left[\hat{\boldsymbol{G}}\right] &= \mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}},\breve{\boldsymbol{P}}_{\boldsymbol{u}}\right] + \mathrm{Cov}\left[\hat{P}_{\mathrm{i}},\hat{P}_{\mathrm{i}}\right]\mathbf{1}\mathbf{1}^t - \left(\mathrm{Cov}\left[\breve{\boldsymbol{P}}_{\boldsymbol{u}},\hat{P}_{\mathrm{i}}\right]\mathbf{1}^t + \mathbf{1}\mathrm{Cov}\left[\hat{P}_{\mathrm{i}},\breve{\boldsymbol{P}}_{\boldsymbol{u}}\right]\right) \\
&= \left(\begin{array}{c:c} \beta_{uu} & \beta_{ul} \\ \hdashline \beta_{lu} & \beta_{ll} \end{array}\right) + \beta_{ii}\mathbf{1}\mathbf{1}^t - \left[\left(\begin{array}{c} \beta_{ui} \\ \hdashline \beta_{li} \end{array}\right)\mathbf{1}^t + \mathbf{1}\left(\begin{array}{c:c} \beta_{iu} & \beta_{il} \end{array}\right)\right].
\end{aligned}$$

This equation allows us to calculate $\mathrm{Cov}\left[\hat{\boldsymbol{b}}_{\mathrm{L}}\right]$ as in (B.4) from the covariance of the complete vector of distance power estimates, $\hat{\boldsymbol{P}}$.

Covariance of $\hat{\boldsymbol{P}}$ for the Different Distance Estimators

Let us reproduce here for convenience the expression of the ML distance estimation (4.11):

$$\hat{d}_{\mathrm{ML}} = d\,10^{\frac{\varepsilon}{10\,n}} = d\exp\left\{\frac{\ln 10}{10n}\varepsilon\right\} = d\,e^{\varepsilon'},\ \ \varepsilon' \sim N(0,\gamma^2), \tag{B.10}$$

where $\gamma = \frac{\sigma \ln 10}{10n}$ as defined in 4.9. The exponentially decaying autocorrelation model for the error vector $\boldsymbol{\varepsilon} = (\varepsilon_1 \dots \varepsilon_{\mathrm{K}})^t$ is

$$\mathrm{Cov}\left[\boldsymbol{\varepsilon}\right] = \sigma_n^2 \mathbf{I} + \sigma_c^2 \boldsymbol{W}. \tag{B.11}$$

where

$$\boldsymbol{W} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,\mathrm{K}} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,\mathrm{K}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{\mathrm{K},1} & w_{\mathrm{K},2} & \cdots & w_{\mathrm{K},\mathrm{K}} \end{pmatrix},\ w_{\mathrm{i,j}} = e^{-l_{\mathrm{i,j}}/l_0},\ l_{\mathrm{i,j}} = \|\vec{\boldsymbol{r}}_{\mathrm{i}} - \vec{\boldsymbol{r}}_{\mathrm{j}}\| \tag{B.12}$$

and recall that $\vec{\boldsymbol{r}}_{\mathrm{i}}$ is the position of the robot at instant i. In Chapter 4 we used Equation (B.10) to calculate the covariance of the ML distance estimator assuming IID errors, which results in $\mathrm{Cov}\left[\boldsymbol{\varepsilon}\right] = \sigma^2 \mathbf{I}$. Instead we now must use (B.11).

Noting that $\varepsilon' = \varepsilon \ln 10/(10n)$, we have that

$$\mathrm{Cov}\left[\boldsymbol{\varepsilon'}\right] = \underbrace{\left(\frac{\sigma_n \ln 10}{10n}\right)^2}_{\gamma_n^2} \mathbf{I} + \underbrace{\left(\frac{\sigma_c \ln 10}{10n}\right)^2}_{\gamma_c^2} \boldsymbol{W} = \gamma_n^2 \mathbf{I} + \gamma_c^2 \boldsymbol{W}. \tag{B.13}$$

From 4.26 we know that the ML estimator of d^m is

$$\hat{P}_{\mathrm{ML}} = \hat{d}^m_{\mathrm{ML}}, \tag{B.14}$$

which, following 4.26, can be expressed as

$$\hat{d}^m_{\mathrm{ML}} = d^m\ e^{m\varepsilon'} = d^m e^{\varepsilon''}.$$

Therefore,

$$\mathrm{Cov}\left[\boldsymbol{\varepsilon''}\right] = \mathrm{Cov}\left[m\boldsymbol{\varepsilon'}\right] = \underbrace{m^2\gamma_n^2}_{\gamma_{mn}^2} \mathbf{I} + \underbrace{m^2\gamma_c^2}_{\gamma_{mc}^2} \boldsymbol{W} = \gamma_{mn}^2 \mathbf{I} + \gamma_{mc}^2 \boldsymbol{W}. \tag{B.15}$$

Let us call $\boldsymbol{q} = \left(e^{\varepsilon''_1}, \dots, e^{\varepsilon''_{\mathrm{K}}}\right)^t$. Because $\boldsymbol{\varepsilon}$ is normally distributed, $\boldsymbol{q}$ is log-normally distributed. The relation between the elements of $\mathrm{Cov}\left[\boldsymbol{\varepsilon''}\right]$ and $\mathrm{Cov}\left[\boldsymbol{q}\right]$ is (see [Žerovnik et al., 2013] for a derivation)

$$\mathrm{Cov}\left[\varepsilon''_i, \varepsilon''_j\right] = \ln\left(\frac{\mathrm{Cov}\left[q_i, q_j\right]}{\mathrm{E}\left[q_i\right]\mathrm{E}\left[q_j\right]} + 1\right), \tag{B.16}$$

and therefore

$$\mathrm{Cov}\left[q_i, q_j\right] = \mathrm{E}\left[q_i\right]\mathrm{E}\left[q_j\right]\left[\exp\left\{\mathrm{Cov}\left[\varepsilon_i'', \varepsilon_j''\right]\right\} - 1\right]. \tag{B.17}$$

Because they are identically and log-normally distributed,

$$\mathrm{E}\left[q_i\right] = \mathrm{E}\left[q_j\right] = e^{\gamma_m^2/2} \equiv \Gamma_m \text{ with } \gamma_m^2 = \gamma_{mn}^2 + \gamma_{mc}^2, \tag{B.18}$$

where we have used (B.15) to compute γ_m. Therefore,

$$\mathrm{Cov}\left[q_i, q_j\right] = \Gamma_m^2\left[\exp\left\{\gamma_{mn}^2\delta_{ij} + \gamma_{mc}^2 w_{ij}\right\} - 1\right], \quad \begin{cases} \delta_{ij} = 1 & : i = j \\ \delta_{ij} = 0 & : i \neq j \end{cases} \tag{B.19}$$

Equivalently, in matrix form we have

$$\boldsymbol{Q} \equiv \mathrm{Cov}\left[\boldsymbol{q}\right] = \Gamma_m^2\left[\exp\left\{\gamma_{mn}^2\mathbf{I} + \gamma_{mc}^2\boldsymbol{W}\right\} - \mathbf{1}_{\mathrm{L}}\mathbf{1}_{\mathrm{L}}^t\right], \tag{B.20}$$

where the exponentiation is element-wise and the vector $\mathbf{1}_{\mathrm{L}}$ has reduced dimension, that is: $\mathbf{1}_{\mathrm{L}} = (1, \ldots, 1)^t$, $\dim(\mathbf{1}_{\mathrm{L}}) = (K-1) \times 1$. We call now $\hat{\boldsymbol{P}}_{\mathrm{ML}} = \left(\hat{P}_{\mathrm{ML}}^1, \ldots, \hat{P}_{\mathrm{ML}}^K\right)^t$, where $\hat{P}_{\mathrm{ML}}^i$ is the ML estimator of d_i^m. The elements of the covariance matrix $\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{ML}}\right]$ are of the form

$$\mathrm{Cov}\left[\hat{d}_{\mathrm{ML}_i}^m, \hat{d}_{\mathrm{ML}_j}^m\right] = d_i^m d_j^m \mathrm{Cov}\left[q_i, q_j\right]. \tag{B.21}$$

Therefore,

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{ML}}\right] = \boldsymbol{D_m} \odot \boldsymbol{Q}, \tag{B.22}$$

where $\odot$ is the Hadamard or element-wise product between matrices and $\boldsymbol{D_m}$ is the matrix of multiplication of the powers of the true distances, defined as

$$\boldsymbol{D_m} = \begin{pmatrix} d_1^m d_1^m & d_1^m d_2^m & \cdots & d_1^m d_{\mathrm{K}}^m \\ d_2^m d_1^m & d_2^m d_2^m & \cdots & d_2^m d_{\mathrm{K}}^m \\ \vdots & \vdots & \ddots & \vdots \\ d_{\mathrm{K}}^m d_1^m & d_{\mathrm{K}}^m d_2^m & \cdots & d_{\mathrm{K}}^m d_{\mathrm{K}}^m \end{pmatrix}, \tag{B.23}$$

From 4.32 we know that the best unbiased estimator of d^m is

$$\hat{P}_{\mathrm{U}} = \frac{\hat{d}_{\mathrm{ML}}^m}{\Gamma_m}. \tag{B.24}$$

Accordingly, the best unbiased estimator of $(d_1^m, \ldots, d_{\mathrm{K}}^m)^t$ is

$$\hat{\boldsymbol{P}}_{\mathrm{U}} = \frac{1}{\Gamma_m}\hat{\boldsymbol{P}}_{\mathrm{ML}}, \tag{B.25}$$

whose covariance is

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{U}}\right] = \frac{1}{\Gamma_m^2}\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{ML}}\right] = \frac{1}{\Gamma_m^2}\boldsymbol{D_m} \odot \boldsymbol{Q}. \tag{B.26}$$

From (4.35) we know that the LMMSE estimator of d^m is

$$\hat{P}_{\mathrm{V}} = \frac{\hat{d}^m_{\mathrm{ML}}}{\Gamma_m^3}. \tag{B.27}$$

Accordingly, the LMMSE estimator of $(d_1^m, \ldots, d_{\mathrm{K}}^m)^t$ is

$$\hat{\boldsymbol{P}}_{\mathrm{V}} = \frac{1}{\Gamma_m^3}\hat{\boldsymbol{P}}_{\mathrm{ML}}, \tag{B.28}$$

whose covariance becomes

$$\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{V}}\right] = \frac{1}{\Gamma_m^6}\mathrm{Cov}\left[\hat{\boldsymbol{P}}_{\mathrm{ML}}\right] = \frac{1}{\Gamma_m^6}\boldsymbol{D_m} \odot \boldsymbol{Q}. \tag{B.29}$$

Appendix C

Appendices for Chapter 7

C.1 Analytical Gradient of the Concentrated Log-Likelihood

In this appendix we derive the expression of the gradient of the concentrated log-likelihood with respect the model parameters (7.21). The following expressions for matrix derivatives shall be useful in our derivation [Rasmussen and Williams, 2006, appx. A.3.1]

$$\frac{\partial}{\partial p}\boldsymbol{K^{-1}} = -\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial p}\boldsymbol{K}^{-1} \tag{C.1}$$

$$\frac{\partial}{\partial p}\ln|\boldsymbol{K}| = \mathrm{Tr}\left(\boldsymbol{K}^{-1}\frac{\partial \boldsymbol{K}}{\partial p}\right), \tag{C.2}$$

where $\frac{\partial \boldsymbol{K}}{\partial p}$ is a matrix of element-wise derivatives wrt. a generic parameter p.

We depart from the original equation:

$$L(\boldsymbol{z};\vec{\boldsymbol{s}},\alpha,l_0) = -\frac{K}{2}(1+\ln 2\pi) - \frac{K}{2}\ln\left(\frac{1}{K}J_\Phi\right) - \frac{1}{2}\ln|\Phi|. \tag{C.3}$$

Its derivative with respect to a generic parameter p is:

$$\frac{\partial L}{\partial p} = -\frac{K}{2J_\Phi}\frac{\partial J_\Phi}{\partial p} - \frac{1}{2}\frac{\partial}{\partial p}\ln|\Phi|, \tag{C.4}$$

and therefore we need to calculate $\frac{\partial J_\Phi}{\partial p}$ and $\frac{\partial}{\partial p}\ln|\Phi|$.

Derivative of J_Φ

First we find an alternative expression of J_Φ which helps us calculate the derivatives as follows:

$$\begin{aligned} J_\Phi &= (\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_\mathrm{m})^t \boldsymbol{\Phi}^{-1} (\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_\mathrm{m}) \\ &= \underbrace{\boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \boldsymbol{z}}_{k_1} - 2 \underbrace{\boldsymbol{z} \boldsymbol{\Phi}^{-1} \boldsymbol{A} \hat{\boldsymbol{\theta}}_\mathrm{m}}_{k_2} + \underbrace{\hat{\boldsymbol{\theta}}_\mathrm{m}^t \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A} \hat{\boldsymbol{\theta}}_\mathrm{m}}_{k_3} \end{aligned}$$

$$k_2 = \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A} \hat{\boldsymbol{\theta}}_\mathrm{m} = \boldsymbol{z}^t \underbrace{\boldsymbol{\Phi}^{-1} \boldsymbol{A} (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^t \boldsymbol{\Phi}^{-1}}_{\boldsymbol{B}} \boldsymbol{z} = \boldsymbol{z}^t \boldsymbol{B} \boldsymbol{z}$$

$$\begin{aligned} k_3 &= \hat{\boldsymbol{\theta}}_\mathrm{m}^t \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A} \hat{\boldsymbol{\theta}}_\mathrm{m} \\ &= [(\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{z}]^t \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A} \boldsymbol{z} (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \\ &= \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A} (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1} \underbrace{\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A} (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1}}_{\mathbf{I}} \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{z} \\ &= \boldsymbol{z}^t \underbrace{\boldsymbol{\Phi}^{-1} \boldsymbol{A} (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^t \boldsymbol{\Phi}^{-1}}_{\boldsymbol{B}} \boldsymbol{z} = \boldsymbol{z}^t \boldsymbol{B} \boldsymbol{z} \end{aligned}$$

$$J_\Phi = k_1 - 2k_2 + k_3 = \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \boldsymbol{z} - 2\boldsymbol{z}^t \boldsymbol{B} \boldsymbol{z} + \boldsymbol{z}^t \boldsymbol{B} \boldsymbol{z} = \boldsymbol{z}^t (\boldsymbol{\Phi}^{-1} - \boldsymbol{B}) \boldsymbol{z}.$$

Using this we have

$$\frac{\partial J_\Phi}{\partial p} = \boldsymbol{z}^t \left(\frac{\partial \boldsymbol{\Phi}^{-1}}{\partial p} - \frac{\partial \boldsymbol{B}}{\partial p} \right) \boldsymbol{z}.$$

Using (C.1) we have

$$\frac{\partial J_\Phi}{\partial p} = -\boldsymbol{z}^t \left(\boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial p} \boldsymbol{\Phi}^{-1} + \frac{\partial \boldsymbol{B}}{\partial p} \right) \boldsymbol{z}. \tag{C.5}$$

We now rewrite $\boldsymbol{B}$ as

$$\boldsymbol{B} = \underbrace{\boldsymbol{\Phi}^{-1} \boldsymbol{A}}_{\boldsymbol{\Psi}} \underbrace{(\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1}}_{\boldsymbol{C}} \underbrace{\boldsymbol{A}^t \boldsymbol{\Phi}^{-1}}_{\boldsymbol{\Psi}^t} = \boldsymbol{\Psi} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t$$

based on which we can calculate the derivative

$$\begin{aligned} \frac{\partial \boldsymbol{B}}{\partial p} &= \frac{\partial}{\partial p} (\boldsymbol{\Psi} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t) = \frac{\partial \boldsymbol{\Psi}}{\partial p} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t + \boldsymbol{\Psi} \frac{\partial}{\partial p} (\boldsymbol{C}^{-1} \boldsymbol{\Psi}^t) \\ &= \frac{\partial \boldsymbol{\Psi}}{\partial p} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t + \boldsymbol{\Psi} \left(\frac{\partial \boldsymbol{C}^{-1}}{\partial p} \boldsymbol{\Psi}^t + \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\Psi}^t}{\partial p} \right) \\ &= \frac{\partial \boldsymbol{\Psi}}{\partial p} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t + \boldsymbol{\Psi} \frac{\partial \boldsymbol{C}^{-1}}{\partial p} \boldsymbol{\Psi}^t + \boldsymbol{\Psi} \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\Psi}^t}{\partial p}. \end{aligned}$$

Using this, we can write (C.5) as:

$$-\frac{\partial J_\Phi}{\partial p} = \boldsymbol{z}^t \left(\boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial p} \boldsymbol{\Phi}^{-1} + \frac{\partial \boldsymbol{\Psi}}{\partial p} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t + \boldsymbol{\Psi} \frac{\partial \boldsymbol{C}^{-1}}{\partial p} \boldsymbol{\Psi}^t + \boldsymbol{\Psi} \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{\Psi}^t}{\partial p} \right) \boldsymbol{z}$$
$$= dJ_{\Phi_1} + dJ_{\Phi_2} + dJ_{\Phi_2}^t + dJ_{\Phi_3}$$

where

$$dJ_{\Phi_1} \equiv \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial p} \boldsymbol{\Phi}^{-1} \boldsymbol{z}$$
$$dJ_{\Phi_2} \equiv \boldsymbol{z}^t \frac{\partial \boldsymbol{\Psi}}{\partial p} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t \boldsymbol{z}$$
$$dJ_{\Phi_3} \equiv \boldsymbol{z}^t \boldsymbol{\Psi} \frac{\partial \boldsymbol{C}^{-1}}{\partial p} \boldsymbol{\Psi}^t \boldsymbol{z}.$$

We shall now particularize the gradient for each parameter.

Gradient for the Position Coordinates

Given that the covariance matrix $\boldsymbol{\Phi}$ depends only on α and l_0, then for the position coordinates we have $\frac{\partial \boldsymbol{\Phi}}{\partial p} = 0$, and therefore $dJ_{\Phi_1} = 0$.

In order to calculate dJ_{Φ_2} we need

$$\frac{\partial \boldsymbol{\Psi}}{\partial x} = \frac{\partial}{\partial x}(\boldsymbol{\Phi}^{-1}\boldsymbol{A}) = \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{A}}{\partial x},$$

with which we can write

$$dJ_{\Phi_2} = \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{A}}{\partial x} \underbrace{\boldsymbol{C}^{-1} \boldsymbol{\Psi}^t \boldsymbol{z}}_{\hat{\boldsymbol{\theta}}_{\mathrm{m}}} = \boldsymbol{u}^t \frac{\partial \boldsymbol{A}}{\partial x} \hat{\boldsymbol{\theta}}_{\mathrm{m}},$$

where we have used $\boldsymbol{C}^{-1}\boldsymbol{\Psi}^t\boldsymbol{z} = (\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\boldsymbol{z} = \hat{\boldsymbol{\theta}}_{\mathrm{m}}$ (see (7.19)) and defined

$$\boldsymbol{u} \equiv \boldsymbol{\Phi}^{-1}\boldsymbol{z}. \tag{C.6}$$

In order to calculate dJ_{Φ_3}, we need $\frac{\partial \boldsymbol{C}^{-1}}{\partial x}$. Using (C.1) we have

$$\frac{\partial \boldsymbol{C}^{-1}}{\partial x} = -\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial x} \boldsymbol{C}^{-1} = -\boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial x} \boldsymbol{\Phi}^{-1} \boldsymbol{A} + \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{A}}{\partial x} \right) \boldsymbol{C}^{-1}$$
$$= -\boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial x} \boldsymbol{\Psi} + \boldsymbol{\Psi}^t \frac{\partial \boldsymbol{A}}{\partial x} \right) \boldsymbol{C}^{-1}.$$

Hence,

$$\begin{aligned} dJ_{\Phi_3} &= -\boldsymbol{z}^t\boldsymbol{\Psi}\boldsymbol{C}^{-1}\left(\frac{\partial \boldsymbol{A}^t}{\partial x}\boldsymbol{\Psi} + \boldsymbol{\Psi}^t\frac{\partial \boldsymbol{A}}{\partial x}\right)\boldsymbol{C}^{-1}\boldsymbol{\Psi}^t\boldsymbol{z} \\ &= -\hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\left(\frac{\partial \boldsymbol{A}^t}{\partial x}\boldsymbol{\Psi} + \boldsymbol{\Psi}^t\frac{\partial \boldsymbol{A}}{\partial x}\right)\hat{\boldsymbol{\theta}}_{\mathrm{m}} = -\hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\frac{\partial \boldsymbol{A}^t}{\partial x}\boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_{\mathrm{m}} - \hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\boldsymbol{\Psi}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}} \\ &= -2\hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\frac{\partial \boldsymbol{A}^t}{\partial x}\boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_{\mathrm{m}}. \end{aligned}$$

The derivative of J_Φ with respect to the x coordinate is then

$$\begin{aligned} -\frac{\partial J_\Phi}{\partial x} &= dJ_{\Phi_2} + dJ^t_{\Phi_2} + dJ_{\Phi_3} = 2\boldsymbol{u}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}} - 2\hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\frac{\partial \boldsymbol{A}^t}{\partial x}\boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_{\mathrm{m}} \\ &= 2\left(\boldsymbol{u}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}} - \hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\frac{\partial \boldsymbol{A}^t}{\partial x}\boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_{\mathrm{m}}\right) \\ &= 2\left(\boldsymbol{u}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}} - \hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\boldsymbol{\Psi}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}}\right) \\ &= 2\left[(\boldsymbol{u}^t - \hat{\boldsymbol{\theta}}^t_{\mathrm{m}}\boldsymbol{\Psi}^t)\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}}\right] \\ &= 2\,\boldsymbol{w}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}}, \end{aligned}$$

where we have defined

$$\begin{aligned} \boldsymbol{v} &\equiv \boldsymbol{\Psi}\hat{\boldsymbol{\theta}}_{\mathrm{m}} = \boldsymbol{\Psi}\boldsymbol{C}^{-1}\boldsymbol{\Psi}^t\boldsymbol{z} \\ \boldsymbol{w} &\equiv \boldsymbol{u} - \boldsymbol{v}. \end{aligned}$$

From (C.4), noting that $\frac{\partial}{\partial p}\ln|\Phi| = 0$, the derivative of the concentrated log-likelihood becomes

$$\frac{\partial L}{\partial x} = -\frac{K}{2J_\Phi}\frac{\partial J_\Phi}{\partial x} = \frac{K}{J_\Phi}\left(\boldsymbol{w}^t\frac{\partial \boldsymbol{A}}{\partial x}\hat{\boldsymbol{\theta}}_{\mathrm{m}}\right).$$

It only remains to calculate

$$\frac{\partial \boldsymbol{A}}{\partial x} = \frac{\partial}{\partial x}\begin{pmatrix}\log(d_1/d_0) & 1\\ \vdots & \vdots\\ \log(d_K/d_0) & 1\end{pmatrix} = \frac{1}{\ln 10}\begin{pmatrix}\frac{(x-r_{x_1})}{d_1^2} & 0\\ \vdots & \vdots\\ \frac{(x-r_{x_K})}{d_K^2} & 0\end{pmatrix},$$

and equivalently,

$$\frac{\partial \boldsymbol{A}}{\partial y} = \frac{1}{\ln 10}\begin{pmatrix}\frac{(y-r_{y_1})}{d_1^2} & 0\\ \vdots & \vdots\\ \frac{(y-r_{y_K})}{d_K^2} & 0\end{pmatrix}.$$

Gradient for the Covariance Model Parameters

For α we have

$$dJ_{\Phi_1} = \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{\Phi}^{-1} \boldsymbol{z} = \boldsymbol{u}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{u}.$$

To calculate dJ_{Φ_2}, we need

$$\frac{\partial \boldsymbol{\Psi}}{\partial \alpha} = \frac{\partial}{\partial \alpha}(\boldsymbol{\Phi}^{-1}\boldsymbol{A}) = -\boldsymbol{\Phi}^{-1}\frac{\partial \boldsymbol{\Phi}}{\partial \alpha}\boldsymbol{\Phi}^{-1}\boldsymbol{A} = -\boldsymbol{\Phi}^{-1}\frac{\partial \boldsymbol{\Phi}}{\partial \alpha}\boldsymbol{\Psi},$$

which results in

$$dJ_{\Phi_2} = -\boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{\Psi} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t \boldsymbol{z} = -\boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{v}.$$

To calculate dJ_{Φ_3}, we need

$$\frac{\partial \boldsymbol{C}}{\partial \alpha} = \boldsymbol{A}^t \frac{\partial \boldsymbol{\Phi}^{-1}}{\partial \alpha} \boldsymbol{A} = -\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{\Phi}^{-1} \boldsymbol{A} = -\boldsymbol{\Psi}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{\Psi},$$

which results in

$$\begin{aligned} dJ_{\Phi_3} &= \boldsymbol{z}^t \boldsymbol{\Psi} \frac{\partial \boldsymbol{C}^{-1}}{\partial p} \boldsymbol{\Psi}^t \boldsymbol{z} = -\boldsymbol{z}^t \boldsymbol{\Psi} \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial p} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t \boldsymbol{z} = \boldsymbol{z}^t \boldsymbol{\Psi} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{\Psi} \boldsymbol{C}^{-1} \boldsymbol{\Psi}^t \boldsymbol{z} \\ &= \boldsymbol{v}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{v}. \end{aligned}$$

Putting dJ_{Φ_1}, dJ_{Φ_2} and dJ_{Φ_2} together, we have

$$\begin{aligned} -\frac{\partial J_\Phi}{\partial \alpha} &= \boldsymbol{u}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{u} - \boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{v} - [\boldsymbol{z}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{v}]^t + \boldsymbol{v}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{v} \\ &= (\boldsymbol{u} - \boldsymbol{v})^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} (\boldsymbol{u} - \boldsymbol{v}) = \boldsymbol{w}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{w}. \end{aligned} \tag{C.7}$$

The derivative of the concentrated log-likelihood (C.4) then becomes

$$-\frac{\partial L}{\partial p} = -\frac{K}{2J_\Phi} \boldsymbol{w}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{w} + \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \right), \tag{C.8}$$

where we have used (C.2) to calculate $\frac{\partial}{\partial \alpha} \ln |\Phi|$. This expression can be further simplified using the cyclic property of matrix traces as follows:

$$\begin{aligned} -\frac{\partial L}{\partial p} &= -\frac{K}{2J_\Phi} \operatorname{Tr}\left(\boldsymbol{w}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \boldsymbol{w} \right) + \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \right) \\ &= -\frac{K}{2J_\Phi} \operatorname{Tr}\left(\boldsymbol{w}\boldsymbol{w}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \right) + \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \right) \\ &= \frac{1}{2} \operatorname{Tr}\left(\boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} - \frac{K}{J_\Phi} \boldsymbol{w}\boldsymbol{w}^t \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \right) \\ &= \frac{1}{2} \operatorname{Tr}\left[\left(\boldsymbol{\Phi}^{-1} - \frac{K}{J_\Phi} \boldsymbol{w}\boldsymbol{w}^t \right) \frac{\partial \boldsymbol{\Phi}}{\partial \alpha} \right]. \end{aligned}$$

We still need to calculate $\frac{\partial \boldsymbol{\Phi}}{\partial \alpha}$. From (7.15) we have

$$\frac{\partial \boldsymbol{\Phi}}{\partial \alpha} = \frac{\partial}{\partial \alpha}[\alpha \mathbf{I} + (1-\alpha)\boldsymbol{W}] = \mathbf{I} - \boldsymbol{W}, \tag{C.9}$$

The expression for the derivative with respect to l_0 is the same, but substituting $\frac{\partial \boldsymbol{\Phi}}{\partial l_0}$ for $\frac{\partial \boldsymbol{\Phi}}{\partial \alpha}$ with

$$\frac{\partial \boldsymbol{\Phi}}{\partial l_0} = \frac{\partial}{\partial l_0}[\alpha \mathbf{I} + (1-\alpha)\boldsymbol{W}] = (1-\alpha)\frac{\partial \boldsymbol{W}}{\partial l_0}.$$

From the definition of $\boldsymbol{W}$ in (6.19), the derivative of the elements of $\boldsymbol{W}$ is:

$$\frac{\partial w_{ij}}{\partial l_0} = \frac{\partial}{\partial l_0} e^{-l_{\mathrm{i,j}}/l_0} = w_{ij}\frac{l_{ij}}{l_0^2}.$$

Therefore,

$$\frac{\partial \boldsymbol{\Phi}}{\partial l_0} = \frac{1-\alpha}{l_0^2}\boldsymbol{W} \odot \boldsymbol{L}, \tag{C.10}$$

where $\odot$ represents the element-wise or Hadamard product between matrices, and

$$\boldsymbol{L} \equiv \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,\mathrm{K}} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,\mathrm{K}} \\ \vdots & \vdots & \ddots & \vdots \\ l_{\mathrm{K},1} & l_{\mathrm{K},2} & \cdots & l_{\mathrm{K,K}}. \end{pmatrix}$$

Gradient for Regularization

Let us call $h(\vec{\boldsymbol{s}}, \boldsymbol{\theta}_{\mathrm{m}}, \sigma, \alpha, l_0)$ a prior distribution over the position and all model parameters and $L_h = \ln(h)$ its natural logarithm. The MAP estimator is then the value that maximizes $L + L_h$, where L is the concentrated log-likelihood of the measurements, which was concentrated substituting $\hat{\boldsymbol{\theta}}_{\mathrm{m}}$ for $\boldsymbol{\theta}_{\mathrm{m}}$ and $\hat{\sigma}$ for σ in the original log-likelihood. In order to retain the low dimensionality of the regularized concentrated log-likelihood, we make the same substitutions in the prior, leading to $h = h(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}, \hat{\sigma}, \alpha, l_0)$. Thus, the same as for L, the log-prior, L_h, depends only on $\vec{\boldsymbol{s}}$, α and l_0. One should, however, be careful when calculating the gradient and apply the chain rule taking into account these substitutions.

Applying the chain rule to calculate $\frac{\partial(L+L_h)}{\partial p}$, where p can be x, y, α or l_0, we have

$$\begin{aligned} \frac{\partial(L+L_h)}{\partial p} &= \frac{\partial L(\vec{\boldsymbol{s}}, \alpha, l_0)}{\partial p} + \frac{\partial L_h(\vec{\boldsymbol{s}}, \hat{\boldsymbol{\theta}}_{\mathrm{m}}, \hat{\sigma}, \alpha, l_0)}{\partial p} \\ &= \frac{\partial L}{\partial p} + \frac{\partial L_h}{\partial p} + \frac{\partial L_h}{\partial \hat{\boldsymbol{\theta}}_{\mathrm{m}}}\frac{\partial \hat{\boldsymbol{\theta}}_{\mathrm{m}}}{\partial p} + \frac{\partial L_h}{\partial \sigma}\frac{\partial \hat{\sigma}}{\partial p}. \end{aligned}$$

The derivatives $\frac{\partial L}{\partial p}$ have just been calculated above, and it only remains to calculate the derivatives of the log-prior. The derivatives $\frac{\partial L_h}{\partial p}$ depend on the particular prior used. We now proceed to calculating $\frac{\partial \hat{\boldsymbol{\theta}}_{\mathrm{m}}}{\partial p}$ and $\frac{\partial \hat{\sigma}}{\partial p}$ for x, y, α and $p = l_0$.

Gradient With Respect to the Position

From (7.19) and using the definition $\boldsymbol{C} \equiv (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})$, we can write

$$\hat{\boldsymbol{\theta}}_{\mathrm{m}} = (\boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{A})^{-1} \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \boldsymbol{z} = \boldsymbol{C}^{-1} \boldsymbol{A}^t \boldsymbol{u}.$$

The gradient then becomes

$$\frac{\partial \hat{\boldsymbol{\theta}}_{\mathrm{m}}}{\partial \vec{\boldsymbol{s}}} = \frac{\partial \boldsymbol{C}^{-1}}{\partial \vec{\boldsymbol{s}}} \boldsymbol{A}^t \boldsymbol{u} + \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{u}.$$

Using

$$\frac{\partial \boldsymbol{C}^{-1}}{\partial \vec{\boldsymbol{s}}} = -\boldsymbol{C}^{-1} \frac{\partial \boldsymbol{C}}{\partial \vec{\boldsymbol{s}}} \boldsymbol{C}^{-1} = -\boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{\Phi}^{-1} \boldsymbol{A} + \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{A}}{\partial \vec{\boldsymbol{s}}} \right) \boldsymbol{C}^{-1},$$

we have

$$\begin{aligned}
\frac{\partial \hat{\boldsymbol{\theta}}_{\mathrm{m}}}{\partial \vec{\boldsymbol{s}}} &= -\boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{\Phi}^{-1} \boldsymbol{A} + \boldsymbol{A}^t \boldsymbol{\Phi}^{-1} \frac{\partial \boldsymbol{A}}{\partial \vec{\boldsymbol{s}}} \right) \underbrace{\boldsymbol{C}^{-1} \boldsymbol{A}^t \boldsymbol{u}}_{\hat{\boldsymbol{\theta}}_{\mathrm{m}}} + \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{u} \\
&= -\boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \underbrace{\boldsymbol{\Phi}^{-1} \boldsymbol{A} \hat{\boldsymbol{\theta}}_{\mathrm{m}}}_{\boldsymbol{v}} + \underbrace{\boldsymbol{A}^t \boldsymbol{\Phi}^{-1}}_{\boldsymbol{\Psi}^t} \frac{\partial \boldsymbol{A}}{\partial \vec{\boldsymbol{s}}} \hat{\boldsymbol{\theta}}_{\mathrm{m}} \right) + \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{u} \\
&= -\boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{v} + \boldsymbol{\Psi}^t \frac{\partial \boldsymbol{A}}{\partial \vec{\boldsymbol{s}}} \hat{\boldsymbol{\theta}}_{\mathrm{m}} \right) + \boldsymbol{C}^{-1} \frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{u} \\
&= \boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \underbrace{(\boldsymbol{u} - \boldsymbol{v})}_{\boldsymbol{w}} - \boldsymbol{\Psi}^t \frac{\partial \boldsymbol{A}}{\partial \vec{\boldsymbol{s}}} \hat{\boldsymbol{\theta}}_{\mathrm{m}} \right) \\
&= \boldsymbol{C}^{-1} \left(\frac{\partial \boldsymbol{A}^t}{\partial \vec{\boldsymbol{s}}} \boldsymbol{w} - \boldsymbol{\Psi}^t \frac{\partial \boldsymbol{A}}{\partial \vec{\boldsymbol{s}}} \hat{\boldsymbol{\theta}}_{\mathrm{m}} \right),
\end{aligned}$$

where we have used the definitions of $\boldsymbol{\Psi}$, $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{w}$ from the previous subsection.

In order to calculate $\frac{\partial \hat{\sigma}}{\partial \vec{\boldsymbol{s}}}$, from (7.20) we obtain

$$\widehat{\sigma^2} = \frac{1}{K} J_\Phi \implies \frac{\partial \hat{\sigma}}{\partial \vec{\boldsymbol{s}}} = \frac{1}{2\sqrt{K J_\Phi}} \frac{\partial J_\Phi}{\partial \vec{\boldsymbol{s}}}. \tag{C.11}$$

The gradient $\frac{\partial J_\Phi}{\partial \vec{\boldsymbol{s}}}$ can be calculated from the gradients calculated in the previous subsections as follows. From (C.4) we see that

$$\frac{\partial L}{\partial x} = -\frac{K}{2J_\Phi}\frac{\partial J_\Phi}{\partial x} \implies \frac{\partial J_\Phi}{\partial x} = -\frac{2J_\Phi}{K}\frac{\partial L}{\partial x},$$

and equivalently for $\frac{\partial L}{\partial y}$. Therefore,

$$\frac{\partial \hat{\sigma}}{\vec{\boldsymbol{s}}} = \frac{1}{2\sqrt{KJ_\Phi}}\frac{2J_\Phi}{K}\frac{\partial L}{\partial \vec{\boldsymbol{s}}} = -\sqrt{\frac{J_\Phi}{K^3}}\frac{\partial L}{\partial \vec{\boldsymbol{s}}}.$$

Gradient With Respect to the Covariance Model Parameters

In order to calculate $\frac{\partial \hat{\boldsymbol{\theta}}_\mathrm{m}}{\partial p}$ for $p = \alpha$ or $p = l_0$, we we start from

$$\hat{\boldsymbol{\theta}}_\mathrm{m} = (\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\boldsymbol{A})^{-1}\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\boldsymbol{z} = \boldsymbol{C}^{-1}\boldsymbol{A}^t\boldsymbol{u}.$$

The gradient then becomes

$$\frac{\partial \hat{\boldsymbol{\theta}}_\mathrm{m}}{\partial p} = \frac{\partial \boldsymbol{C}^{-1}}{p}\boldsymbol{A}^t\boldsymbol{u} + \boldsymbol{C}^{-1}\boldsymbol{A}^t\frac{\partial \boldsymbol{\Phi}^{-1}}{\partial p}\boldsymbol{z}.$$

Using

$$\frac{\partial \boldsymbol{C}^{-1}}{p} = -\boldsymbol{C}^{-1}\frac{\partial \boldsymbol{C}}{\partial p}\boldsymbol{C}^{-1} = -\boldsymbol{C}^{-1}\left(\boldsymbol{A}^t\frac{\partial \boldsymbol{\Phi}^{-1}}{\partial p}\boldsymbol{A}\right)\boldsymbol{C}^{-1},$$

we have

$$\begin{aligned}
\frac{\partial \hat{\boldsymbol{\theta}}_\mathrm{m}}{\partial p} &= -\boldsymbol{C}^{-1}\left(\boldsymbol{A}^t\frac{\partial \boldsymbol{\Phi}^{-1}}{\partial p}\boldsymbol{A}\right)\boldsymbol{C}^{-1}\boldsymbol{A}^t\boldsymbol{u} + \boldsymbol{C}^{-1}\boldsymbol{A}^t\frac{\partial \boldsymbol{\Phi}^{-1}}{\partial p}\boldsymbol{z} \\
&= \boldsymbol{C}^{-1}\boldsymbol{A}^t\frac{\partial \boldsymbol{\Phi}^{-1}}{\partial p}\left(\boldsymbol{z} - \boldsymbol{A}\underbrace{\boldsymbol{C}^{-1}\boldsymbol{A}^t\boldsymbol{u}}_{\hat{\boldsymbol{\theta}}_\mathrm{m}}\right) = -\boldsymbol{C}^{-1}\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}\frac{\partial \boldsymbol{\Phi}}{\partial p}\boldsymbol{\Phi}^{-1}(\boldsymbol{z} - \boldsymbol{A}\hat{\boldsymbol{\theta}}_\mathrm{m}) \\
&= -\boldsymbol{C}^{-1}\underbrace{\boldsymbol{A}^t\boldsymbol{\Phi}^{-1}}_{\boldsymbol{\Psi}^t}\frac{\partial \boldsymbol{\Phi}}{\partial p}\underbrace{(\boldsymbol{\Phi}^{-1}\boldsymbol{z} - \boldsymbol{\Phi}^{-1}\boldsymbol{A}\hat{\boldsymbol{\theta}}_\mathrm{m})}_{\boldsymbol{u}-\boldsymbol{v}=\boldsymbol{w}} \\
&= -\boldsymbol{C}^{-1}\boldsymbol{\Psi}^t\frac{\partial \boldsymbol{\Phi}}{\partial p}\boldsymbol{w}.
\end{aligned}$$

To calculate $\frac{\partial \hat{\sigma}}{\partial p}$, using (C.11) for p instead of $\vec{\boldsymbol{s}}$ and using (C.7) we have

$$\frac{\partial \hat{\sigma}}{\partial p} = \frac{1}{2\sqrt{KJ_\Phi}}\frac{\partial J_\Phi}{\partial p} = -\frac{1}{2\sqrt{KJ_\Phi}}\boldsymbol{w}^t\frac{\partial \boldsymbol{\Phi}}{\partial p}\boldsymbol{w},$$

and the derivatives $\frac{\partial \boldsymbol{\Phi}}{\partial p}$ have already been calculated in (C.9) and (C.10) for $p = \alpha$ and $p = l_0$, respectively.

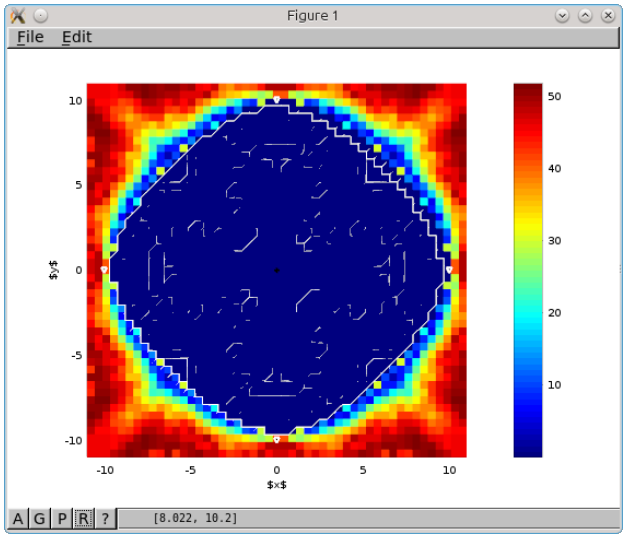


Figure 1
File Edit
10
5
0
-5
-10
y
-10
-5
0
5
10
x
50
40
30
20
10
A G P R ?
[8.022, 10.2]

2AKR

This thesis investigates how to effectively use mobile reference devices to help locating the nodes of a wireless sensor network (WSN) using received signal strength (RSS) measurements. In particular, a robot capable of self-localization and mapping is used as a mobile beacon. The aim is to produce accurate, precise, practical and computationally efficient localization methods that can be used in real ad-hoc deployments without prior training data, such as emergency and rescue scenarios. Although using RSS for localization is a well established research field, how to do it fully exploiting the rich spatial sampling capabilities offered by mobile reference devices is still largely unexplored. The studies presented here benefit from this richness in order to obtain better RSS models and position estimators. The proposed localization methods are validated using real measurements collected in three different indoor environments with a demonstrated sub-meter accuracy in all of them.

Aalto-DD 56/2016

ISBN 978-952-60-6725-4 (printed)
ISBN 978-952-60-6726-1 (pdf)
ISSN-L 1799-4934
ISSN 1799-4934 (printed)
ISSN 1799-4942 (pdf)

Aalto University
School of Electrical Engineering
Department of Electrical Engineering and Automation
www.aalto.fi

BUSINESS + ECONOMY

ART + DESIGN + ARCHITECTURE

SCIENCE + TECHNOLOGY

CROSSOVER

DOCTORAL DISSERTATIONS